430-TP-009-001

Data Acquisition Request Prototype Results

Technical paper - Not intended for formal review or Government approval.

September 1996

Prepared Under Contract NAS5-60000

RESPONSIBLE ENGINEER

John Ujhazy /s/	9/30/96
John Ujhazy EOSDIS Core System Project	Date

SUBMITTED BY

Mary S. Armstrong /s/	9/30/96
Mary Armstrong EOSDIS Core System Project	Date

Hughes Information Technology Systems
Upper Marlboro, Maryland

This page intentionally left blank.

Abstract

The Earth Observing System Data and Information System (EOSDIS) Core System (ECS) involves the collection and distribution of data from space and ground based measurement systems to provide the scientific basis for understanding of global change. The Data Acquisition Request (DAR) prototype undertook the design and development of a prototype for the submission of data acquisition requests for data from the Advanced Spaceborne Thermal Emission and Reflection Radiometer (ASTER) instrument, which is a part of the AM-1 mission. This document details the prototype design and development effort.

Keywords: ASTER, DAR, GDS, AM-1, ECS, EOSDIS

This page intentionally left blank.

Contents

1. Introduction

2. DAR API Interface

3. DAR GUI Prototype

4. DAR Analysis and Design Materials

5. Development Notes

Figures

Tables

Appendices

1. Introduction

1.1 Purpose and Scope

This white paper documents the results of the Data Acquisition Request (DAR) Prototype. The document is intended to provide a single point of reference for information generated in the course of developing of the prototype This document can be used to understand the engineering details of the DAR prototype, and as a resource for further development of DAR functionality. The document focus is on the Phase 2 DAR prototype, which was a following-on to Phase 1 DAR prototyping efforts.

It is suggested that readers unfamiliar with the operational concepts associated with the ASTER instrument consult appropriate references listed in section 1.4 for background materials.

1.2 Overview of the DAR Prototypes

The DAR prototyping effort was implemented in two phases. The Phase 1 prototype focus was on developing Graphical User Interfaces (GUIs) to explore the DAR user interface functionality. After several iterations of Phase 1 GUI development, it became obvious that the Phase 1 prototyping effort was not structured to support the iterative activities necessary for successful prototyping of the DAR tool. The Phase 1 prototype was completed, and a Phase 2 follow-on prototype was outlined with expanded objectives and well-defined iterations. Iterations included customer reviews and comments, and occurred on roughly monthly cycles. The details of both prototypes, including monthly status reports, are available in the Prototyping and Studies Progress Report for the ECS Project.

The Phase 2 DAR prototype focus changed during the course of the prototype from GUI development to analysis consisting of workflows, and scenario development. From early feedback on the Phase 2 GUI, it became obvious that the DAR tool was more complex than originally conceived, and that a comprehensive workflow analysis was necessary to analyze this complexity and provide the basis for further design and development work. Consequently, the latter portion of the prototype work saw a shift of focus from GUI prototyping to development of workflows to describe in detail the components of the DAR tool. The prototype was completed with work being done on two complementary tracks. A coded GUI was used to explore implementations and provide additional feedback to developers on areas identified with development risks. The second effort at analysis/design provided workflows and scenarios for DAR functionality.

1.3 Document Organization

This paper is organized as follows:

Section 1. Introduction, provides descriptive information about this document.

Section 2 DAR Interface, describes the Application Programming Interface (API) which will provided for communication of the DAR tool with the ASTER Ground Data System in Japan.

Section 3, GUI Prototype Description, provides an as-built specification of the developed GUI.

Section 4, DAR Workflows, documents the workflows developed for the prototype.

Section 5, Development Notes, documents other prototyping activities, and provides recommendation for further development of DAR functionality, including identification of design issues.

Three appendices provide information related to the DAR prototype.

Appendix A. DAR Prototype GUIs, contains screen dumps of the as-built prototype, along with descriptions of screen functions.

Appendix B. Timeline Development Notes, provides a description of the capabilities of the Delphi Timeline libraries, and the as-built timeline software.

Appendix C, Guidelines for Graphical User Interface (GUI) Development Using the Human-Machine Interface (HMI) Methodology, is a description of the workflow methodology used in the DAR prototype analysis.

Questions regarding technical information contained within this paper should be addressed to the following ECS contacts:

John Ujhazy, Prototyping Manager

Ph: 301-925-0468

Email: jujhazy@eos.hitc.com

Questions concerning distribution or control of this document should be addressed to:

Data Management Office

The ECS Project Office

Hughes Information Technology Corporation

1616 McCormick Dr.

Upper Marlboro, MD 20785

1.4 References

ASTER Functional Requirements for Mission Operations, ASTER science team, version .7, March 20, 1995.

EOSDIS Core System Project, Prototyping and Studies Progress Report for the ECS Project, 318-CD-000-021.

Cohen, R. (1996). *XAR Parameter Change Request from US Science Team to Japan Science Team*. Jet Propulsion Lab.

EOSDIS Core System Project, Prototyping and Studies Progress Report for the ECS Project, 318-CD-000-021.

Jet Propulsion Lab. (1994) *ASTER DAR Generation and Processing Concepts.* (JPL D-11504). Prepared by the Instrument Operations Support Task Design Team.

Mitsubishi Electric Co., Ltd. (1996). *ASTER-GDS IMS DAR Client API List*. (API-LIST-1).

Mitsubishi Electric Corporation (1996). *DAR Input Parameter List*. Presentation to the ECS-ASTER GDS IMS Interface Meeting in Landover, Md on June 4-7, 1996.

Mitsubishi Electric Corporation (1996). *GDS IMS DAR Client Overview*. Presentation to the ECS-ASTER GDS IMS Interface Meeting in Landover, Md on June 4-7, 1996.

(US) ASTER Science Team. (March 20, 1995). (DRAFT) *Functional Requirements for Mission Operations.* (Version .7).

This page intentionally left blank.

2. DAR API Interface

2.1 Description

The Scheduling Database for the ASTER instrument is part of the ASTER Ground Data System (GDS), which is located in Japan. DAR-related communications between ECS and the GDS is accomplished through a GDS provided API, which will be hosted at the DAR Gateway server at the Eros Data Center (EDC). The DAR API as provided by the GDS is written in C. The DAR API will be provided with a object wrapper at the DAR Gateway by the Infrastructure Development Group (IDG), so that the DAR client will need to access only Distributed Object Framework (DOF) classes. The wrapper for the API will provide standard DOF-related capabilities to the API (security, naming, etc.), but will not extend the DAR functionality itself; this is limited by the functionality provided by the ASTER GDS through the API set.

2.2 Interface Details

2.2.1 API Structures

The DAR API consists of six C language calls. As noted above, the object wrapper for the API wrapper will not change the functionality supported by the API set, so for prototyping purposes the API set was used to provide a baseline of DAR parameters and valids. The current baseline of the API set was provided to ECS at the ECS GDS Interface Coordination Meeting of July 22-26, 1996. Later versions of the ASTER Interface Control Document should include the API specification.

2.2.2 DAR Valids

DAR valids are used for submission and search of DARs, and the component parts of DARs. The current baseline is for the GDS to control the configuration of DAR valids, and to provide them to the ECS. The current baseline additionally calls for the valids to be provided by the GDS in GDS format, with ECS converting them to ECS format. The DAR valids are expected to change infrequently, if ever, so valids maintenance is not expected to be a factor. For purposes of the DAR prototype, the DAR API was examined to isolate DAR parameters and valids.

2.2.3 API Scenarios

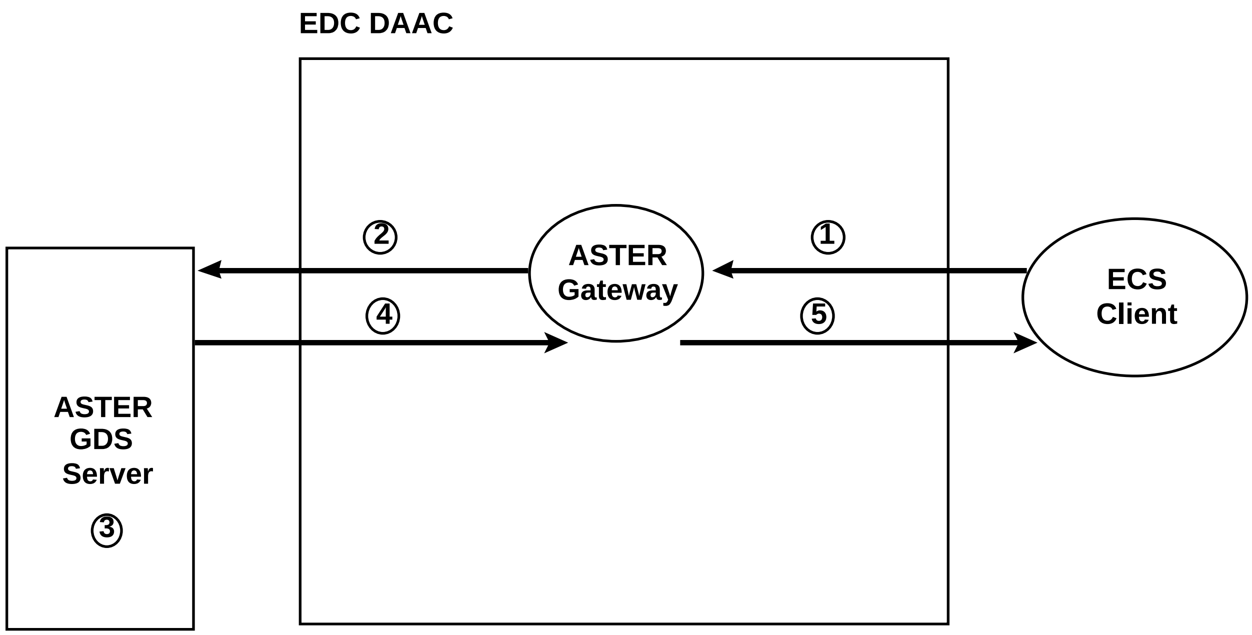
Three scenarios were developed illustrate the use of the API set by the DAR tool. These scenarios while not exercising the entire API set, cover the most critical DAR functions.

Figure 2-1 provides a dataflow for the submission of a DAR by an ECS user to the ASTER GDS. Table 2-1 details each of the steps enumerated in the dataflow.

Figure 2-2 provides a dataflow for the statusing of a existing DAR by an ECS user. Table 2-2 details each of the steps enumerated in the dataflow.

Figure 2-3 provides a dataflow for query of observed scenes by an ECS user. Table 2-2 details each of the steps enumerated in the dataflow.

Column three in the following tables documents technical questions about the interface which are currently outstanding.

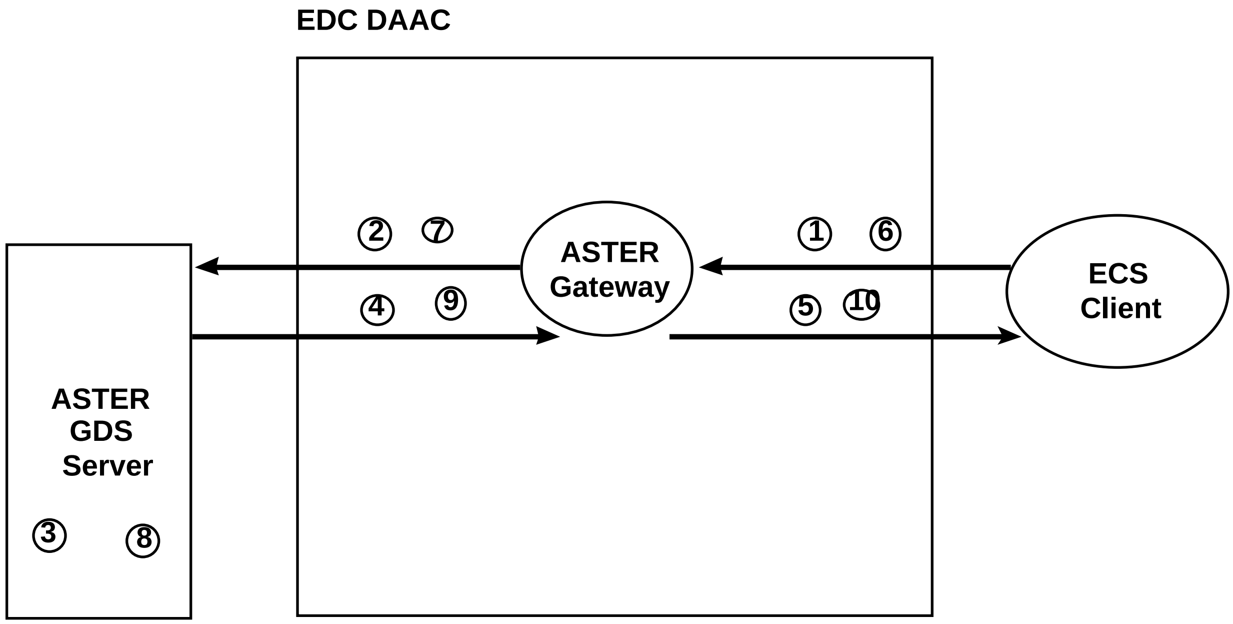


1. DAR Submit (ECS DAR Client to ASTER Gtway Server)
2. DAR Submit (ASTER Gtway Server to ASTER GDS
3. DAR storage in local holdings
4. DAR ID (ASTER GDS to Gt way Server)
5. DAR ID (ASTER Gtway Server to ECS DAR Client

Figure 2-1. Dataflow for DAR Submission

Table 2-1. Step Details for DAR Submission Dataflow

Step Description	Message & Format	Comments/Issues/Questions
1. DAR Client submits DAR data to the ASTER Gateway	ECS Internal	DAR Client does not check ECS inventory for observed scenes
2. ASTER Gateway isolates DAR parameters and makes a call to the DAR API	submitDAR (*xarDataStream, *DarID)	
3. ASTER GDS Stores DAR information locally	NA	
4. ASTER GDS returns a DAR ID	DarID parameter returned in pointer to above	Turnaround time of 10 seconds for the API call Issue of management of US and Japanese allocations for acquisition bandwidth still open
5. ASTER Gateway extracts returned DAR ID from return and returns it to the ECS DAR Client	ECS Internal	Client saves DAR ID locally for future use

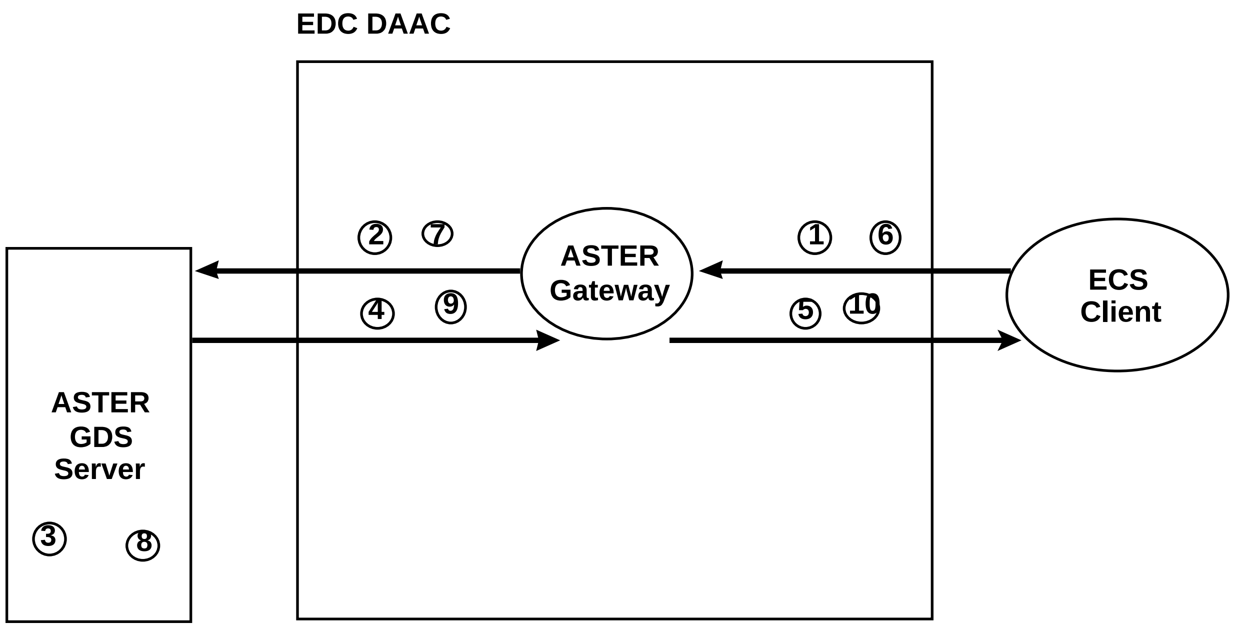


1. DAR Status Request (ECS DAR Client to ASTER Gtway Server)
2. DAR Status Request (ASTER Gtway Server to ASTER GDS)
3. DAR Status Processing
4. DAR Status Result (ASTER GDS to Gtway)
5. DAR Status Result (Gtway to DAR Client)
6. subDAR Status Request (ECS DAR Client to ASTER Gtway Server
7. subDAR Status Request (ASTER Gtway Server to ASTER GDS)
8. subDAR Status Processing
9. subDAR Status Result (ASTER GDS to Gtway)
10. subDAR Status Result (Gtway to DAR Client)

Figure 2-2. Dataflow for DAR Status

Table 2-2 Step Details for DAR Status Dataflow

Step Description	Message & Format	Comments/Issues/Questions
1. DAR Client requests DAR status from the ASTER Gateway	ECS Internal	Assume DAR Client has DAR ID
2. ASTER Gateway makes a call to the DAR API	getXARStatus(*searchStream, *resultStream) where *searchStream->xar_id = <user's GDS assigned DAR ID>	Search Stream can contain temporal, spacial other constraints
3. ASTER GDS processes DAR status query	NA	
4. ASTER GDS returns a DAR ID	*resultStream contains DAR status in return from call	Turnaround time for the API call needs estimation
5. ASTER Gateway returnsDAR status to the ECS DAR Client for display	ECS Internal	How does the GDS DAR client display this status information?
6. DAR Client requests subXAR status from Gateway	ECS Internal	
7. ASTER Gateway makes call to the ASTER API	getSubXARStatus(*searchSubxarStream, *subxarstream) where *searchSubxarStream->xar_id = <user's parent XAR ID>	Other temporal, spacial, instrument constraints can be added.
8. ASTER GDS processes status query	NA	
9. ASTER GDS returns query result to ASTER Gateway	*subxarstream contains subxars query status in return from call	ASTER Science Team requested status by scene quad, geolocated. Is this going to be supported? Turnaround time for the API call needs estimation
10. ASTER Gateway returns result to DAR Client	ECS Internal	DAR Client draws AOI and plots scenes



1. DAR Status Request (ECS DAR Client to ASTER Gtway Server)
2. DAR Status Request (ASTER Gtway Server to ASTER GDS)
3. DAR Status Processing
4. DAR Status Result (ASTER GDS to Gtway)
5. DAR Status Result (Gtway to DAR Client)
6. subDAR Status Request (ECS DAR Client to ASTER Gtway Server
7. subDAR Status Request (ASTER Gtway Server to ASTER GDS)
8. subDAR Status Processing
9. subDAR Status Result (ASTER GDS to Gtway)
10. subDAR Status Result (Gtway to DAR Client)

<11 … N Repeat Steps 6 thru 10 for each XAR>

Figure 2-3. Dataflow for Query on Observed Scenes

Table 2-3 Step Details for Query on Observed Scenes (1 of 2)

Step Description	Message & Format	Comments/Issues/Questions
1. DAR Client requests list of DARs from the ASTER Gateway which match certain criteria	ECS Internal	
2. ASTER Gateway makes a call to the DAR API	getXARStatus(*searchStream, *resultStream) where *searchStream->num_polygon = <number of points in area of query> *searchStream->lst_point-_long = <long of first point in area of query> *searchStream->lst_point-_lat = <lat of first point in area of query> … *searchStream->start_time = <start time of interest to user > *searchStream->end_time = <end time of interest to user >	Other constraints can be added to query as required For DAR type, a 0x03 = All is needed Are the *searchStream parameters for upper and lower lat and long now necessary given that a polygonal AOQ is specified?
3. ASTER IMS processes DAR query	NA	How are the start and end time and polygonal coordinates used to query for results i.e., does the result set contain the ids of all XARs which intersect these parameters?
4. ASTER GDS returns result set to ASTER Gtway	*resultStream contains list of DARs, STARs, ETRs and other metadata meeting search return in return from call	Turnaround time for the API call needs estimation
5. ASTER Gateway returnsDAR status to the ECS DAR Client for display	ECS Internal	
6. DAR Client requests subXAR status from Gateway	ECS Internal	

Table 2-3 Step Details for Query on Observed Scenes (2 of 2)

Step Description	Message & Format	Comments/Issues/Questions
7. ASTER Gateway makes call to the ASTER API	getSubXARStatus(*searchSubxarStream, *subxarstream) where *searchSubxarStream->start_time = <start time specified in DAR search> *searchSubxarStream->end_time = <end time specified in DAR search>	Other temporal, spacial, instrument constraints can be added. Can steps 1 to 5 be omitted such that a constrained on all subxars will return the observed scenes?
8. ASTER GDS processes status query	NA	
9. ASTER GDS returns query result to ASTER Gateway	*subxarstream contains subxars from the XAR which match criteria	ASTER Science Team requested status by scene quad, geolocated. Is this going to be supported?
10. ASTER Gateway returns result to DAR Client	ECS Internal	DAR Client draws AOI and plots scenes
11. Steps 6 thru 10 repeated for each XAR in results set	As specified above	

3. DAR GUI Prototype

3.1 GUI Prototype Description

The DAR GUI, as prototyped, consists of a interface that functions in three modes of operation: DAR Submission Mode, DAR Status Mode, and DAR Query Mode. The layout of the DAR tool was implemented to be consistent with the X-motif version of the Earth Science Search Tool (ESST). The individual buttons which serve to access submenus change according to which mode is selected. Areas on the main screen provide for listing of parameters selected in the submenus.

The DAR tool consists of three modules. The main module is text and icon based, and provides a user the central point for changing modes and accessing other DAR tool windows. A map module provides a user the capability to visualize geographic data used in the DAR tool, including orbital information and DAR coverage. A timeline module provides the capability to timeline multiple DAR related events, such as instrument acquisition schedules and user requested acquisitions.

On startup, the DAR tool comes up in DAR submission mode. A set of buttons controls switching the tool between modes. In each mode, buttons provide access to submenus which are used to accept inputs for various parameters applicable to the mode in which the tool is in.

Refer to Appendix A for screen dumps of the GUI screens, with detailed descriptions of the each screen's operation.

3.2 As-built Prototype Mapping to Requirements

The ASTER Functional Requirements for Mission Operations (FRMO) was used as a starting point for determining the functionality of the coded prototype. The following table shows the FRMO requirements and how they apply to the as-built prototype.

Table 3-1. As-built Requirements for DAR Prototype (1 of 11)

Number	Text	Prototype
3.2.1	Information Management System (Common)	
3.2.1.1	General	
3.2.1.1-1	The IMS shall provide the user the ability to electronically and interactively create and submit DARs.	Yes
3.2.1.1-2	The IMS shall provide the user the capability to electronically and interactively track the status of all xARs in the planning and scheduling data base.	Yes (status is mocked up)
3.2.1.1-3	The IMS user shall have simultaneous access to at least three types of information displays. They are:	
a)	Geographic displays	Yes
b)	Schedule timeline displays	Yes
c)	Acquisition request and observation information displays in tabular and text form	Yes
3.2.1.1-4	The information in all three displays shall be automatically (under user control) updated when the information changes in one of the displays.	Partial. Some but not all displays coordinated
3.2.1.1-5	The user shall be able to display the results of queries on the three types of displays.	Yes
3.2.1.1-6	Map and timeline display updates (such as zoom or scroll operations) shall be completed within a maximum of 15 (TBR) seconds.	Yes
3.2.1.1-7	The IMS shall support an overlapping multi-window type of display, user-controlled via a pointing device such as a mouse.	Yes
3.2.1.1-8	The display within a given window shall be scrollable and printable by the user.	scrollable = yes , printable = no
3.2.1.1-9	All objects on timeline displays and map overlays shall be selectable with a pointing device.	Yes
3.2.1.2 Request Submittal		
3.2.1.2-1	The IMS shall be able to accept the following mandatory DAR content items:	
	Requester name and address	Yes
	Requester authorization identifier	Yes
	Area(s) of interest	Yes
	Observation time window(s)	Yes
	Spectral channels (telescopes)	Yes
	Investigation classification	Yes
	Scientific objective	Yes
	Illumination requirements (day/night)	Yes

Table 3-1. As-built Requirements for DAR Prototype (2 of 11)

Number	Text	Prototype
3.2.1.2-2	The IMS shall be able to accept the following optional DAR content items:	
	Specific time and location of observation (e.g. orbit number, time relative to node crossing).	Yes
	Area of Interest coverage requirements (e.g. sampled, contiguous within each swath, minimum acceptable amount of coverage)	Yes
	Pointing and pointing angle constraints	Yes
	Sun angle	Yes
	Gain settings	Yes
	Desired target condition (e.g. snow cover, cloud cover)	Yes
	Acceptable data quality (e.g. cloud cover)	Yes
	Seasonality requirements	Yes
	Calibration requirements	Yes
	Observation coordination requirements	Yes
	Direct downlink option (DDL)	Yes
	Fragmentability	Yes
Unless otherwise specified, all datatakes must be contiguous within an individual swath		
	An indication of and justification for an unexpected urgent event (TOO)	Yes
3.2.1.2-3	Each DAR parameter entry field shall indicate whether it is a required entry or whether it is optional.	No
3.2.1.2-4	Each optional DAR entry field shall have appropriate default values.	Yes
3.2.1.2-5	Syntax and parameter value limit checking shall be performed during DAR parameter entry, and after each entry is completed.	Partial: Checking is not done on all parameters
3.2.1.2-6	The IMS shall have the capability for the user to define an area of interest and desired coverage by entering the following:	
a)	point target — center lat/lon and radius	yes
b)	area target — polygon containing area or center point, shape, and size	yes
c)	sampled regions — region polygon and sample duration or region point lat/lon/proximity and sample duration	N/A; predefined shapes not identified in workflows
d)	irregular linked — contour points lat/lon and extent distance or contour point definition and extent distance	N/A; contour sensitivity not identified in workflows

Table 3-1. As-built Requirements for DAR Prototype (3 of11)

Number	Text	Prototype
3.2.1.2-7	The IMS shall allow geographic location data to be interactively entered by user selection of point locations and polygon boundary points by a pointing device such as a mouse. A screen-displayed background map shall be provided for visual reference.	Yes
3.2.1.2-8	Checks shall be made of the area of interest specification upon completion by the user to verify that the specified area is valid.	N/A; no AOI validation checks identified.
3.2.1.2-9	The IMS shall make consistency checks of all DAR parameters upon submittal by the user.	Partial; not all parameters checked
3.2.1.2-10	After the DAR is submitted by the user, an indication of the likelihood of scheduling shall be provided.	N/A; DAR client will support scheduling status through status query.
	Such as, how many observations are possible with the requirements specified in the DAR, and when the first observation may be scheduled.	
3.2.1.2-11	The IMS user shall have the opportunity to generate a Data Product Request (DPR) associated with a DAR at the same time the DAR is submitted.	No
3.2.1.2-12	The IMS shall receive, from the ICC, planning and scheduling information which is necessary to construct geographic, schedule timeline or text displays.	Yes (Read from database)
3.2.1.3-1	The IMS shall provide the user with the capability to query the planning and scheduling data base to select xARs or groups of xARS which satisfy user-defined search criteria relating to xAR entry parameters as defined in requirements 3.2.1.2-1 and 3.2.1.2	Partial. Selected parameters used for query, and results read from database
3.2.1.3-2	The IMS shall provide the user with the capability to query the planning and scheduling data base to select xARs or groups of xARS by specifying:	
a)	acquisition window dates	No, since DAR query baseline was unknown when DAR tool was coded.
b)	area of interest name	No, since DAR query baseline was unknown when DAR tool was coded
c)	investigation class	Yes
d)	investigator	No, since DAR query baseline was unknown when DAR tool was coded

Table 3-1. As-built Requirements for DAR Prototype (4 of 11)

Number	Text	Prototype
b)	Investigation name	No, since DAR query baseline was unknown when DAR tool was coded
e)	xAR Identification Number	No, since DAR query baseline was unknown when DAR tool was coded
3.2.1.3-3	The IMS shall provide the user the capability to select xARs or groups of xARs by:	
a)	selecting area(s) of interest on a geographical display	
b)	selecting observations or observation opportunities on either a geographical display or timeline schedule	Yes
c)	specifying a time interval on a timeline or geographic display	Yes on timeline, not on map
3.2.1.3-4	The IMS shall provide the user with the capability to display the information associated with the xAR(s) identified by a query. This information shall include both user-controllable parameters (see 3.2.1.2-1 and 3.2.1.2-2) and additional data produced by	
a)	xAR Identification Number	No, since DAR query baseline was unknown when DAR tool was coded
b)	Orbits in which observations can be acquired	No, since DAR query baseline was unknown when DAR tool was coded
c)	Observation opportunities	No, since DAR query baseline was unknown when DAR tool was coded
d)	Actual observations	No, since DAR query baseline was unknown when DAR tool was coded
	(pointers to archived data sets)	No, since DAR query baseline was unknown when DAR tool was coded
e)	Pointing-angle of orbits in which observations can be acquired	
f)	Actual sun-angle of observation (or opportunity)	No, since DAR query baseline was unknown when DAR tool was coded

Table 3-1. As-built Requirements for DAR Prototype (5 of 11)

Number	Text	Prototype
g)	Area of interest size	No, since DAR query baseline was unknown when DAR tool was coded
h)	Observation history, indicating success/failure of past observation attempts	No, since DAR query baseline was unknown when DAR tool was coded
i)	Observation potential, indicating future number of observation opportunities	No, since DAR query baseline was unknown when DAR tool was coded
3.2.1.3-5	The IMS shall provide the capability to sort xARs based on user specification of one or more fields from the xAR information.	Yes
3.2.1.3-6	The IMS shall acquire xAR status from the ICC and display it to the user. xAR status shall include:	
a)	xAR state	Yes
b)	completion status	Yes
c)	processing status	
	- Level 0 results	No
	- Level 1 results	No
3.2.1.4 Map Display		
3.2.1.4-1	The IMS shall provide the user a geographic map display that consists of background data used for reference and overlay data (graphics and annotation) that illustrate the spatial characteristics of xARs, observation opportunities, or actual observations.	yes
3.2.1.4-2	The IMS map display shall provide the following types of geographic data sets (in a variety of projections) for background reference:	
a)	a physiographic map providing terrain and surface-cover features globally at approximately 1-km resolution	No
	1-km data sources (e.g. AVHRR or Digital Chart of the World) may be combined as required to create this map.	
b)	physical and cultural features such as:	
	land/oceans	Yes
	major lakes	Yes
	major rivers	Yes
	mountain ranges (in topographic form)	No
	volcanoes	No
	major highways and railroads	No
	urban and built-up areas	No
c)	political boundaries	Yes

Table 3-1. As-built Requirements for DAR Prototype (6 of 11)

Number	Text	Prototype
d)	A digital elevation model at no less than 10km horizontal resolution	No
	Such as, ETOPO5 5-minute (10km) gridded elevations/bathymetry for the world	
e)	higher resolution (100m-1 km) digital elevation models of selected areas in the U.S. and Japan	No
f)	earthquake epicenters since 1980	No
g)	time and space-averaged land vegetation index(es)	No
h)	seasonal snow/ice cover (average/maximum)	No
i)	actual snow cover (within the past TBD days)	No
j)	cloud cover probabilities (climatology)	No
3.2.1.4-3	The IMS map display shall provide the following types of geographic overlays registered to the background data sets (see 3.2.1.4-2):	
a)	ground tracks with orbit number, time, orbit relative time	
b)	areas of interest with:	No
c)	terminator crossings	No
d)	equator and other major latitude crossings	No
e)	instrument constraints (applicable to map)	
-	field of view	No
-	pointing ranges	No
-	nominal view swaths	No
-	non-nominal view swaths	No
-	maximum observation duration	No
f)	begin and end times of instrument activities plotted on orbital ground tracks	Yes
-	pointing	Yes
-	Telescope mode (state)	Yes
g)	outlines of observation opportunities	Yes
h)	outlines of scheduled observations	Yes
i)	outlines of successful observations (with data quality indicators)	
j)	sun angle	Yes
3.2.1.4-4	The IMS shall have the capability to add or subtract overlays or backgrounds to the map region on a geographic display, utilizing the same map projections and at the currently selected zoom factor.	Yes
3.2.1.4-5	The IMS shall have the capability to zoom in and out of areas of a geographical display up to a TBD range and scale.	Yes
3.2.1.4-6	The IMS shall provide a zoom capability based on selection of, at a minimum, any of the following:	
a)	specified lat/lon center point or corner points	No

Table 3-1. As-built Requirements for DAR Prototype (7 of 11)

Number	Text	Prototype
b)	a point selected on a displayed map using a pointing device, and a zoom factor	Yes
c)	drawing an arbitrary rectangle on the screen	Yes
c)	the selection of an observation outline on a map	No
d)	the specification of an observation outline by name	
e)	the selection of an area of interest on a map	Yes
f)	the specification of an area of interest by name	No
g)	specification of a place-name	No
3.2.1.4-7	The IMS shall provide the capability to pan across areas on a geographical display in user-controllable increments.	Yes
3.2.1.4-8	The IMS shall provide the capability to jump directly to user-specified features including areas of interest, observations, observation opportunities and place-names.	no
3.2.1.4.9	The IMS shall zoom or pan all geographic map data currently displayed, including all overlays and backgrounds.	yes
3.2.1.4.10	The IMS shall have the capability to display the lat/lon of any point on a geographical display selected by the user's pointing device.	No
3.2.1.4.11	The IMS shall be able to calculate and display the size (in square km) of any user-specified area (polygonal or center-plus-radius) on a geographical display.	no
3.2.1.4.12	The IMS shall be able to calculate and display the size (in square km) of any user-selected area of interest or observation on a geographical display.	no
3.2.1.4.13	The IMS shall be able to calculate and display the shortest great-circle distance (in km) between any two user-specified points. The points may be specified by selection of two points on a geographic display, or by entering two lat/lon locations, or some	no
3.2.1.4.14	The IMS shall have the capability to display on a geographical map display the areal extent of an observation outline from an observation specified in a schedule timeline.	No
3.2.1.4.15	The IMS shall have the capability to display on a geographical map display the areal extent of an observation opportunity corresponding to a particular orbit, look angle, and that falls within a user-specified time period.	No
3.2.1.4.16	The IMS shall filter observation data displayed on a geographical map display based on xAR/observation selections made by the user.	Yes
3.2.1.4.17	The IMS shall display orbital ground tracks, view-swaths and orbit times using the best possible ephemeris values.	Partial
3.2.1.4.18	The IMS shall display all possible view swaths from which a selected area of interest can be seen, and all observation opportunities for the area of interest, given the pointing extent available to the instrument.	N/A. This capability not identified in data flows

Table 3-1. As-built Requirements for DAR Prototype (8 of 11)

Number	Text	Prototype
3.2.1.5 Schedule Timeline Display		
3.2.1.5-1	The IMS shall receive schedule data from the ICC for instrument and spacecraft activities.	Yes (dummy data used)
3.2.1.5-1	The IMS shall have the capability to display instrument and AM-1 spacecraft activities and related information chronologically on a schedule timeline.	Yes (dummy data used)
3.2.1.5-2	For any selected orbits and/or time period of the input schedule data, the IMS shall have the capability to display any user-defined combination of the following, at a minimum, on a timeline schedule:	
a)	observations scheduled	No, but capability exists in timeline library.
b)	pointing activities (for each telescope)	No, but capability exists in timeline library.
c)	mode changes (for each telescope)	No, but capability exists in timeline library.
d)	calibration activities (for each telescope)	No, but capability exists in timeline library.
e)	sun-angle (from the ground, at the center of the view-swath for the pointing-angle used at each point in the timeline)	No, but capability exists in timeline library.
f)	terminator crossings	No, but capability exists in timeline library.
g)	equator crossings	No, but capability exists in timeline library.
h)	land areas that fall within the instrument viewing range	
i)	orbit number in a 16-day repeat cycle	No, but capability exists in timeline library.
j)	16-day repeat cycle number	No, but capability exists in timeline library.
k)	GMT of user-selected position on the timeline (using pointing device)	No, but capability exists in timeline library.
l)	activities of instruments other than ASTER, requiring coordination with ASTER	No, but capability exists in timeline library.

Table 3-1. As-built Requirements for DAR Prototype (9 of 11)

Number	Text	Prototype
m)	activities of the spacecraft, potentially requiring coordination (e.g., SSR volume, TDRSS contacts, downlink data volume)	No, but capability exists in timeline library.
n)	local solar time for each pointing angle view swath	No, but capability exists in timeline library.
3.2.1.5-3	For scheduled observations or observation opportunities, the IMS shall display the following information at a minimum, on a timeline schedule display:	
a)	start and stop time in GMT and orbit-relative times	No, but capability exists in timeline library.
b)	Telescopes used	No, but capability exists in timeline library.
c)	Pointing angle (nominal view swath, or off-nominal pointing angle)	
d)	Gain setting of each telescope	No, but capability exists in timeline library.
e)	Band selection of the VNIR subsystem	No, but capability exists in timeline library.
f)	Sampled region indicator	No, but capability exists in timeline library.
g)	Fragmentation parameters	No, but capability exists in timeline library.
h)	xAR ID	No, but capability exists in timeline library.
3.2.1.5-4	The IMS shall filter observation data displayed on a schedule timeline based on xAR/observation selections made by the user.	yes
3.2.1.5-5	The IMS shall provide an indication when a subset of the observations are displayed (as a result of filtering) within a given schedule window.	yes
3.2.1.5-6	The IMS shall calculate and display the absolute (GMT) time difference between any two user-selected points on the timeline schedule display.	no
3.2.1.5-7	The IMS shall calculate and display the absolute (GMT) time difference between the end of one event and the beginning of another event, as selected by the user on a timeline schedule display.	no
3.2.1.5-8	The scale of the timeline display shall be controllable by the user.	yes

Table 3-1. As-built Requirements for DAR Prototype (10 of 11)

Number	Text	Prototype
3.2.1.5-9	The IMS shall allow the user to continuously scroll through the timeline display such that the actual timeline being displayed is one time-segment (window) within a longer timeline.	yes
3.2.1.6 Miscellaneous		
3.2.1.6-1	The IMS shall provide a general information text display which contains ASTER instrument and science information. This information shall include such things as:	N/A for prototype; Guide Access through EP7 software
a)	science products which may be generated	N/A for prototype; Guide Access through EP7 software
b)	science investigation classes from the LTIP and their respectively priorities	N/A for prototype; Guide Access through EP7 software
c)	individual telescope characteristics	N/A for prototype; Guide Access through EP7 software
-	spectral bands	N/A for prototype; Guide Access through EP7 software
-	ground resolution	N/A for prototype; Guide Access through EP7 software
-	FOV	N/A for prototype; Guide Access through EP7 software
-	pointing capabilities	N/A for prototype; Guide Access through EP7 software
-	scanning modes	N/A for prototype; Guide Access through EP7 software
-	gain settings	N/A for prototype; Guide Access through EP7 software

Table 3-1. As-built Requirements for DAR Prototype (11 of 11)

Number	Text	Prototype
-	data volume generated	N/A for prototype; Guide Access through EP7 software
-	duty cycle limitations	N/A for prototype; Guide Access through EP7 software
3.2.1.6-2	The IMS shall provide sample DAR entries with associated map displays.	No
3.2.1.6-3	A sample map display shall be available for each investigation class, which includes:	No
a)	area of interest types and how they need to be specified	
b)	telescope and bands (may be done by color-coding)	
3.2.1.6-4	The IMS shall provide direct human support to users for assisting with data acquisition issues via voice communications.	N/A to prototype
3.2.1.6-5	The IMS capabilities shall be supported on, at least, the following types of operating systems:	
a) Microsoft Windows		No
b) Apple Macintosh OS		No
c) UNIX		Sun only

4. DAR Analysis and Design Materials

4.1 DAR Workflows

User interaction with computer software proceeds in sequences or steps that collectively define the workflow for using the software tool to produce a given product (e.g., results of a database query, report on the occurrence of an event, transmission of a user-created electronic mail message). A methodology, known as the Human-Machine Interface (HMI) methodology has been selected by ECS to analyze the workflows of science users in interacting with ECS Client software. The result of the application of the HMI methodology is the development of a Graphical User Interface (GUI) that users find intuitive and efficient to use in performing their science work. One of the key components of the HMI methodology, workflow analysis, was a major focus of the present prototyping activity. This section discusses the use, purpose, and results of the workflow analysis for the DAR prototype, beginning with a brief summary of the HMI methodology and applicable guidelines for its use in ECS developments.

4.1.1 Application of ECS Guidelines on the Human-Machine Interface (HMI) Methodology

Through its efforts to integrate human factors engineering into the GUI design and development process, ECS has provided guidelines for GUI development using the HMI Methodology. This eighteen page guideline is attached to this report as Appendix A. The HMI methodology prescribes four major activities, including: (1) data input analysis, (2) workflow analysis, (3) preparation of screen layout drawing packages (aka paper prototyping), and (4) GUI screen development using Builder Xcessory as the ECS GUI builder tool for all custom Motif GUIs. The preferred approach is to perform each activity in the sequence identified, however, the successful application of the HMI methodology depends on the analysts ability to tailor the methodology to match the situation.

In the case of the DAR prototyping effort, there were two departures from the preferred HMI development sequence. First, step 1. Data Input Analysis, was a truncated effort due to the fact that the DAR was undergoing rapid prototyping at the time that the application of the HMI methodology was initiated for the DAR effort. Data Input Analysis depends upon the availability of at least a preliminary object model from which to extract objects to incorporate into data input matrices that identify those objects with which users interact and/or manipulate through a GUI environment. In order to launch the HMI effort, pseudo-objects were identified by extracting technical information from the ASTER documentation produced by the Jet Propulsion Lab (JPL) and Mitsubishi Electric Corporation. Of particular importance were the DAR Input Parameter List and the (DRAFT) Functional requirements for mission operations.

Consequently, the DAR workflow analysis became the primary focus of the HMI effort. The success of the workflow analysis was linked directly to ECS staff access to scientists and engineers from the JPL, representing their accumulated understanding of the broader ASTER community as well as their insights into the entire ASTER development program. In this case, 'success' is defined as the degree to which workflow diagrams were prepared that captured an accurate depiction of the user interactions needs of the ASTER science community in interacting with the DAR Tool.

The second departure from the preferred sequence in the HMI methodology, was the preparation of DAR GUI prototypes prior to the completion of the DAR workflow diagrams. This pragmatic decision to accelerate the DAR GUI development through rapid prototyping was to resolve a number of technical risk factors associated with the use of COTS tools to support DAR mapping and timeline requirements—design considerations, whose resolution, were fundamental to the success of the prototyping effort. The outcome of this effort is primarily in Section 5. Continued Development, and will not be repeated here. However, it is noteworthy to state that a detailed human factors evaluation was conducted at various stages in the development of the evolving prototype. These evaluations addressed areas of compatibility of the rapid prototype with the new DAR workflows as well as specific human engineering deficiencies that will be addressed in later stages of DAR GUI development.

The process of bringing the DAR GUI prototypes into compliance with extant HFE standards as defined in the *ECS User Interface Style Guide* and to accommodate the DAR workflow diagrams to the extent practicable is discussed in Section 5. Continued Development.

4.1.2 DAR Top-Level Functions

The top-level functions for the DAR that formed the basis for the preparation of the DAR data input matrices and the DAR workflow diagrams are:

1.0 Create/Submit a XAR

- 1.1 Initiate create/edit mode for a new or previous XAR
- 1.2 Identify areas of interest (polygon)
- 1.3 Identify observation timing
- 1.4 Identify instrument configuration
- 1.5 Identify viewing geometry
- 1.6 Identify data quality
- 1.7 Identify miscellaneous
- 1.8 Create/submit a DPR with a XAR
- 1.9 Save local copy [of work in progress] of XAR configuration parameters

1.10 Estimate resources required to produce ASTER product in accordance with XAR

1.11 Revise XAR parameters based on excessive resource requirements

1.12 Submit XAR

2.0 Create/Submit a XAR Query to the Japanese ASTER database

2.1 Create a query of XARs for the Japanese ASTER database using the ESST

2.2 Submit query to the Japanese ASTER database using the ESST

2.3 Browse/filter/sort the summary results displayed on the ESST summary results screen [XAR by XAR]

2.4 Decide to view XAR details or refine query?

2.5 Refine query

2.6 Submit refined query

2.7 Browse/filter/sort query results using the ESST summary results display

2.8 Track XAR status (accepted, rejected, scheduling, partially observed, re-scheduled, completed, failed, canceled, or suspended)

2.9 Decide to create a XAR, track XAR status or observed scenes, order ASTER products, or view schedules?

2.10 Track XAR status by viewing observed scenes and/or valid data for any XAR

2.11 Observe observed scenes or overlapping AOIs that match specified spatial, temporal, and other search criteria

2.12 Order existing ASTER products using the ESST

2.13 View temporal parameter(s) in conjunction with overlay of ASTER mission schedule(s)

4.1.3 Preparation of the DAR Data Input Matrices and Workflow Diagrams

Two preliminary data input matrices (DIM) were prepared for each of the two key DAR functions. The data input matrix for DAR Function 1.0, Create/Submit a XAR, is shown in Table 4-1. The primary source of pseudo-object data was the *DAR Input Parameters List* (Mitsubishi Electric Corporation, 1996). Each pseudo-object is organized around each of the ten operator functions required to create/submit a XAR. Data elements, display type, and the type of operator interaction required for a user to interact or modify the pseudo-objects are shown in the remaining tables. Again, this information should be treated as preliminary, pending the revisions necessary to replace the current pseudo-objects with the actual object and object elements defined in the as-built object model data for the DAR.

Table 4-1. Data Input Matrix of Operator Interactions with the DAR Client Tool to Create/Submit a XAR (1 of 4)

Operator Function	Object (Pseudo)	Data Element (Object Attribute)	Display	Operator Interactions
1.1 Initiate create/edit mode for a new or previous XAR	DAR_ID_Data	-XAR ID (*logged by system*) -XAR Type ([DAR]/STAR/ETR/other) -XAR Title -sensor type ([ASTER], other?) -date & time of original XAR submission (YYYY/MM/DD/hh:mm:ss) -date & time of latest XAR submission (YYYY/MM/DD/hh:mm:ss) -requester ID (ECS User ID) -investigation classification (volcano, cloud, glacier, calibration, other) -ground campaign (Yes/[No]) -scientific objective (text) -implementation urgency ([Normal]/urgent) -SSSG cancellation ([Active]/Suspend) -user cancellation ([Active]/Suspend)	DAR Form (D) D “ “ “ “ “ “ “ “ “ “ “ “	Display Set Enter Set Display Display Display Set Set Enter Set Set Set
1.2 Identify Areas of Interest (AOI) polygon	AOI_polygon	-coverage method ([Normal]/Sampled) -minimum sample length [60 km] -maximum sample length (km) -number of samples required -coverage amount required (0 - 100%; [100%]) -latitude for @ point 1..n (deg) -longitude for @ point 1..n (deg) -allow cross-track fragmentation ([Yes]/No) -require full duration observations across AOI ([Yes]/No)	Map (M) M “ “ “ “ “ “ “	Set Set/Enter Set/Enter Set/Enter Set Select Select Set Set
	Vu_swath_ data Orbit_data	-start time -stop time -view swath ID -view swath angle limits -groundtrack -view swath -orbits ([ascending]/descending)	“ “ “ “ “ “ “	Set/Enter Set/Enter Select/Display Select/Display Display Display Set/Display

	Zoom_Pan	-zoom %	“	Set/Enter
		-pan (coordinates)	“	Select/Enter

Table 4-1. Data Input Matrix of Operator Interactions with the DAR Client Tool to Create/Submit a XAR (2 of 4)

Operator Function	**Object (Pseudo)**	**Data Element (Object Attribute)**	**Display**	**Operator Interactions**
	MAP_Char	-physiographic terrain/surface map	“	Display
		-land/oceans	“	Display
		-major lakes	“	Display
		-major rivers	“	Display
		-topographically-displayed mountain ranges	“	Display
		-volcanoes	“	Display
		-major highways and railroads	“	Display
		-urban and built-up areas	“	Display
		-political boundaries	“	Display
		-digital elevation	“	Display
		-high resolution digital elevation	“	Display
		-earthquake epicenters (since 1980)	“	Display
		-time/space averaged land vegetation indexes	“	Display
		-seasonal snow/ice cover	“	Display
		-actual snow cover	“	Display
		-cloud cover probabilities	“	Display
1.3 Identify observation timing	OB_Timing	-multi-observations? (Yes/[No])	Timeline (T)	Set
		-XAR lifetime start (YYY/MM/DD/hh:mm:ss)	T	Set/Enter
		-XAR lifetime end (YYY/MM/DD/hh:mm:ss)	“	Set/Enter
		-repeats? (Yes/[No])	“	Set
		-repeat interval (RI) duration (DDD/hh:mm:ss; [XAR lifetime = end - start])	“	Set/Enter
		-Number of repeat intervals [Max # of RIs that fit XAR lifetime]	“	Set/Enter
		-acquisition window (AW) duration (DD/hh:mm:ss; [RI duration])	“	Set/Enter
		-maximize number of revisits in AW (Yes/[No])	“	Set
		-number of revisits in AW [1]	“	Set
		-use specific observation time (Yes/[No])	“	Set
		-specific orbit number	“	Enter/Set
		-specific observation start time (mm:ss)	“	Enter/Set
		-specific observation duration (mm:ss)	“	Enter/Set

Table 4-1. Data Input Matrix of Operator Interactions with the DAR Client Tool to Create/Submit a XAR (3 of 4)

Operator Function	**Object (Pseudo)**	**Data Element (Object Attribute)**	**Display**	**Operator Interactions**
1.4 Identify instrument configuration	INST_Config	-instrument mode (Full, VNIR, VNIR Stereo, SWIR & TIR, TIR)	D	Set
		-VNIR band 1 gain setting (high gain, normal gain, low gain 1; [tbd])	"	Set
		-VNIR band 2 gain setting (high gain, normal gain, low gain 1; [tbd])	"	Set
		-VNIR band 3 gain setting (high gain, normal gain, low gain 1; [tbd])	"	Set
		-SWIR band 4 gain setting (high gain, normal gain, low gain 1, low gain 2; [tbd])	"	Set
		-SWIR band 5 gain setting (high gain, normal gain, low gain 1, low gain 2; [tbd])	"	Set
		-SWIR band 6 gain setting (high gain, normal gain, low gain 1, low gain 2; [tbd])	"	Set
		-SWIR band 7 gain setting (high gain, normal gain, low gain 1, low gain 2; [tbd])	"	Set
		-SWIR band 8 gain setting (high gain, normal gain, low gain 1, low gain 2; [tbd])	"	Set
1.5 Identify Viewing Geometry	VU_Geom	-day/night (Day, Night, Day and Night, Day or Night)	M	Set
		-minimum sun angle [0 deg]	"	Set/Enter
		-maximum sun angle [90 deg]	"	Set/Enter
		-minimum look angle (0 - 24 deg VNIR or V3B/V3N mode; 0 - 8.54 deg all others; [0 deg])	"	Set/Enter
		-maximum look angle (0 - 24 deg VNIR or V3B/V3N mode; 0 - 8.54 deg all others; [8.54 deg])	"	Set/Enter
		-use specified look angle (Yes/[No])	"	Set
		-specific look angle (deg + or -, relative to nadir; [0])	"	Set/Enter
		-use specific view swath (Yes/[No])	"	Set
		-specific view swath	"	Set/Enter
1.6 Identify Data Quality	CLUD_CVR	-avoid clouds ([Yes]/No)	"	Set
		-maximum cloud cover (0 - 100%; [tbd])	"	Set/Enter
1.7 Identify miscellaneous	MISC_TOPics	-expedited data (Yes/[No])	T	Set
		-direct downlink (Yes/[No])	"	Set
		-comments (text)	D	Enter

1.8 Create/submit a DPR with a XAR	DPR_wXAR	tbd	Pull-down menu (on same screen as the Submit XAR function)	Select

Table 4-1. Data Input Matrix of Operator Interactions with the DAR Client Tool to Create/Submit a XAR (4 of 4)

Operator Function	Object (Pseudo)	Data Element (Object Attribute)	Display	Operator Interactions
1.9 Save local copy [of work in progress] of XAR configuration parameters	n/a	n/a	Dialog	Select/Enter
1.10 Estimate resources required to produce ASTER product in accordance with XAR	RES_EST	Resource Estimate Algorithm (Defined as: Total AOI Coverage = Area x (Total # Repeats) or: Total AOI Coverage = Area x (# Repeats) [Scene area (3600 km^2)]	on same screen as the Submit XAR function	Select
1.11 Revise XAR parameters based on excessive resource requirements	n/a	n/a	n/a	n/a
1.12 Submit XAR	SUBM_XAR	-Submit? -acknowledge receipt (XAR ID, other?)	D “	Select Display

The data input matrix for DAR Function 2.0, Create/Submit a XAR Query, is shown in Table 4-2. The primary source of pseudo-object data was the *ASTER-GDS IMS DAR Client API list* (Mitsubishi Electric Corporation, 1996). In a number of ways, this data input matrix is less mature than that prepared for DAR Function 1.0. Given the state of ASTER development activity, it appeared prudent to reference the latest relevant documentation as the source of pseudo-object data, rather than generate pseudo-object data that could potentially prove confusing later. Again, this matrix needs to be updated to reflect the actual object and object elements contained in the as-built object model for the DAR.

Table 4-2. Data Input Matrix of Operator Interactions with the DAR Client Tool to Create/Submit a XAR Query (1 of 3)

Operator Function	Object (Pseudo)	Data Element (Object Attribute)	Display	Operator Interactions
2.1 Create a query of XARs for the Japanese ASTER database using the ESST	XAR_Query Param_Query : Spatial Param_Query : Temporal Param_Query : Instruments, Etc.	-Data Elements listed in Table 2 (searchStream) of API (Item No. 1) -Data Elements listed in Table 2 (searchStream) of API (Items No. 55 through (56+2N+6)) -Data Elements listed in Table 2 (searchStream) of API (Items No. 40-48) -Data Elements listed in Table 2 (searchStream) of API (Items No. 2-39; except for user-entered text fields)	Query Form (Q) Map (M) Timeline (T) Q	Enter Select/Enter Select/Set/ Enter Enter/Set
2.2 Submit query to the Japanese ASTER database using the ESST	n/a	n/a	n/a	n/a
2.3 Browse/ filter/sort the summary results displayed on the ESST summary results screen	n/a	n/a	n/a	n/a
2.4 Decide to view XAR details or refine query?	n/a	n/a	n/a	n/a
2.5 Refine query	see 2.1			
2.6 Submit refined query	n/a	n/a	n/a	n/a
2.7 Browse/ filter/sort query results using the ESST summary results display	n/a	n/a	n/a	n/a

Table 4-2. Data Input Matrix of Operator Interactions with the DAR Client Tool to Create/Submit a XAR Query (2 of 3)

Operator Function	Object (Pseudo)	Data Element (Object Attribute)	Display	Operator Interactions
2.8 Track XAR status	XAR_Status	XAR_Status (Item #12 of Table 3 (resultStream) of ASTER API)	Q	Select
2.9 Decide to create a XAR, track XAR status or observed scenes, order ASTER products, or view schedules?	n/a	n/a	n/a	n/a
2.10 Track XAR status by viewing observed scenes and/or valid data for own XAR	OB_Scenes	subXAR_Status (items on Table 6 (searchSubxarStream) of ASTER API)	Q/M/T	Enter/Set/Select
2.11 Observe observed scenes or overlapping AOIs that match specified spatial, temporal, and other search criteria	Broad_search	subXAR_Status (items on Table 6 (searchSubxarStream) of ASTER API)	Q/M/T	Enter/Set/Select
2.12 Order existing ASTER products using the ESST	see ESST object model	see ESST object model	ESST Order Screen	Select

Table 4-2. Data Input Matrix of Operator Interactions with the DAR Client Tool to Create/Submit a XAR Query (3 of 3)

Operator Function	Object (Pseudo)	Data Element (Object Attribute)	Display	Operator Interactions
2.13 View temporal parameter(s) in conjunction with overlay of ASTER mission schedule(s)	Param_Query : Temporal	Data Elements listed in Table 5 (scheduleStream) of ASTERAPI	Timeline (T)	Select/Set/ Enter

As stated the data input matrices provided one basis for the preparation of DAR workflow diagrams. However, given the preliminary nature of the matrices, the primary source of workflow data were extracted from interviews with JPL scientists and engineers working with the ASTER science community and from the extant ASTER design documents listed in the References. The sections that follow present the workflow diagrams for the two key DAR functions.

4.1.3.1 Workflow Diagrams for DAR Function 1.0: Create/Submit a XAR

Figure 4-1 presents the top-level workflow diagram for DAR Function 1.0. Each of the remaining figures (4-2 through 4-13) present the workflows necessary for end-users to execute the DAR Functions 1.1 through 1.10 using the DAR Client Tool.

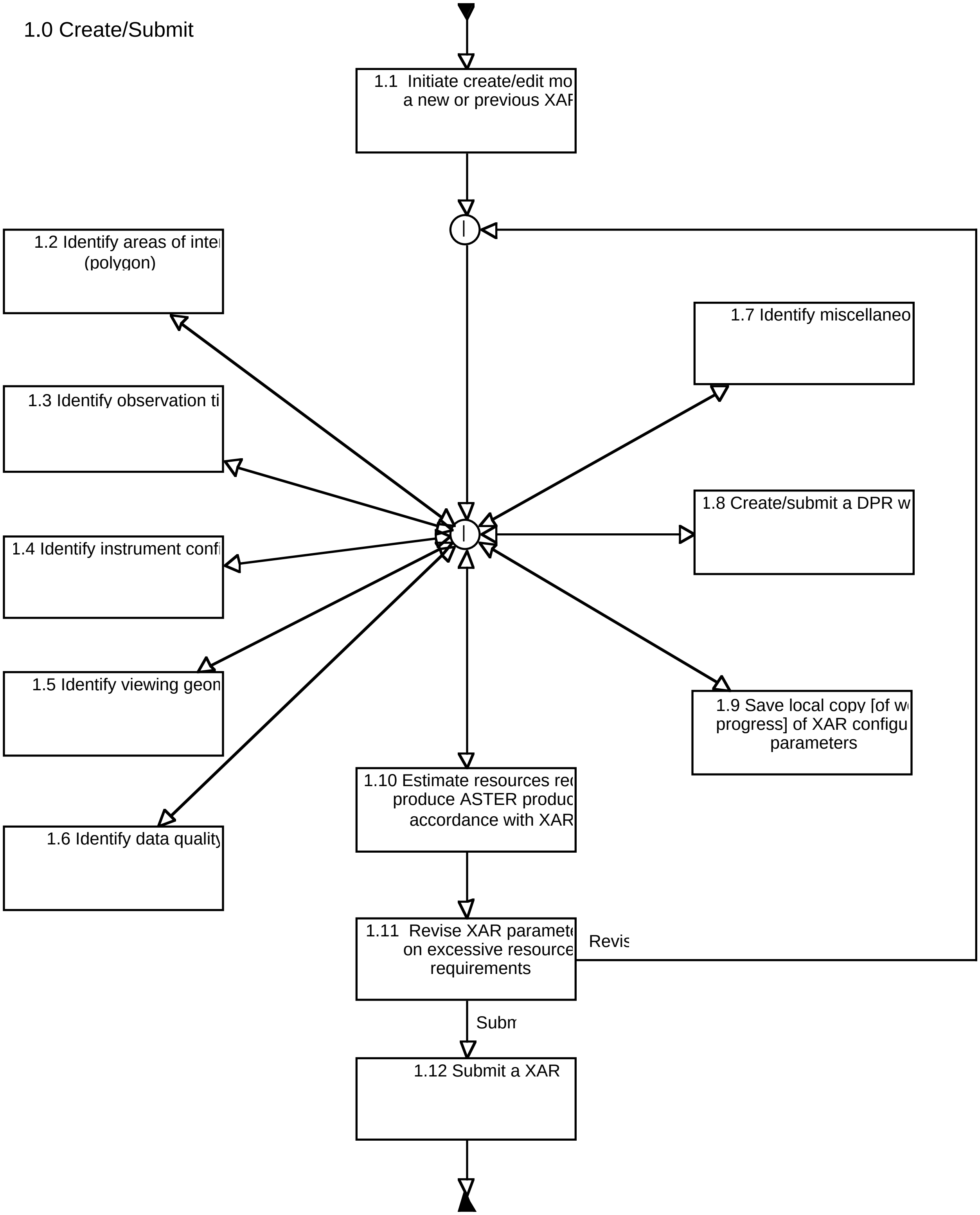


Figure 4-1. Top-Level Workflow Diagram for DAR Function 1.0

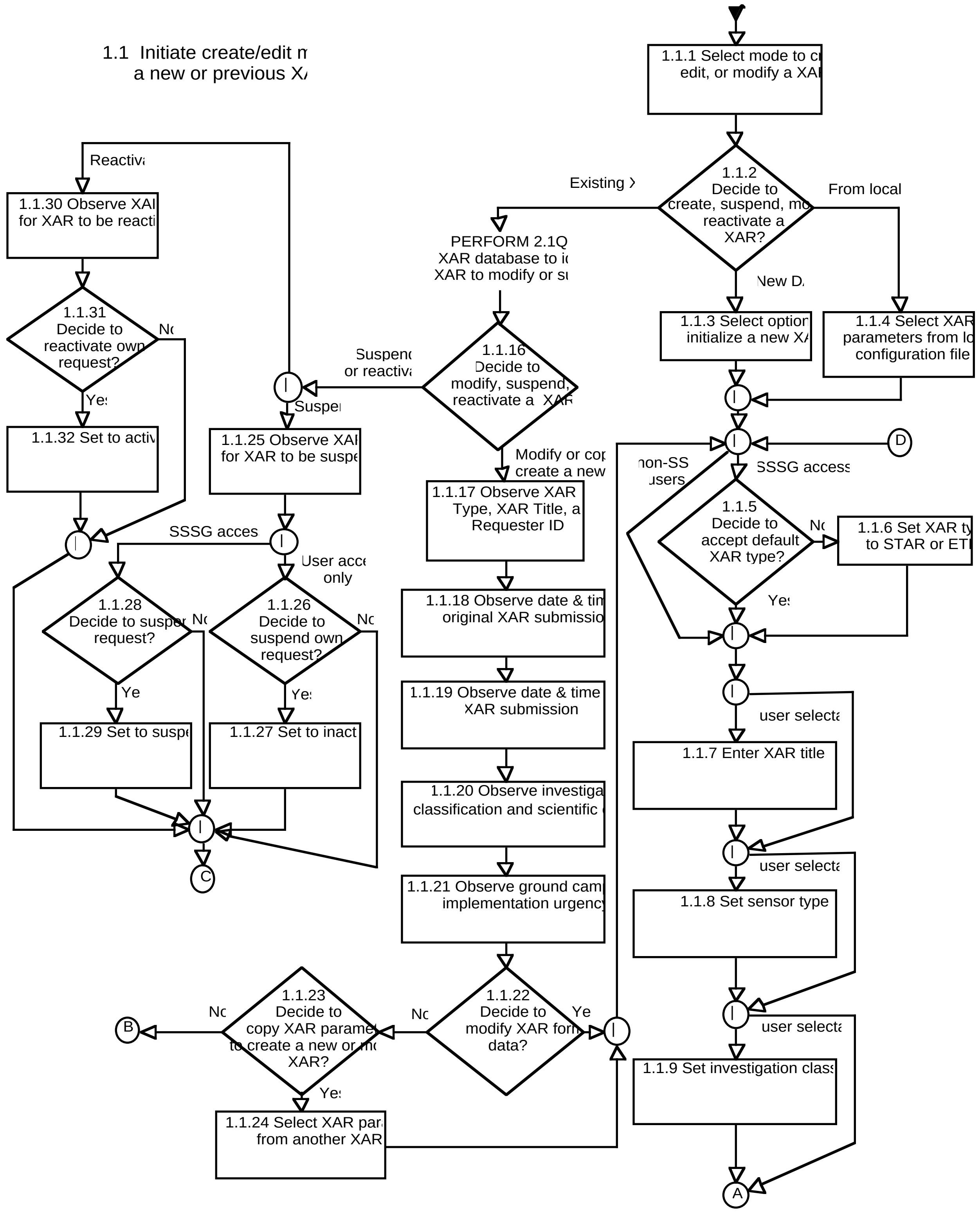


Figure 4-2. Detailed Workflow for DAR Function 1.1 (1 of 2)

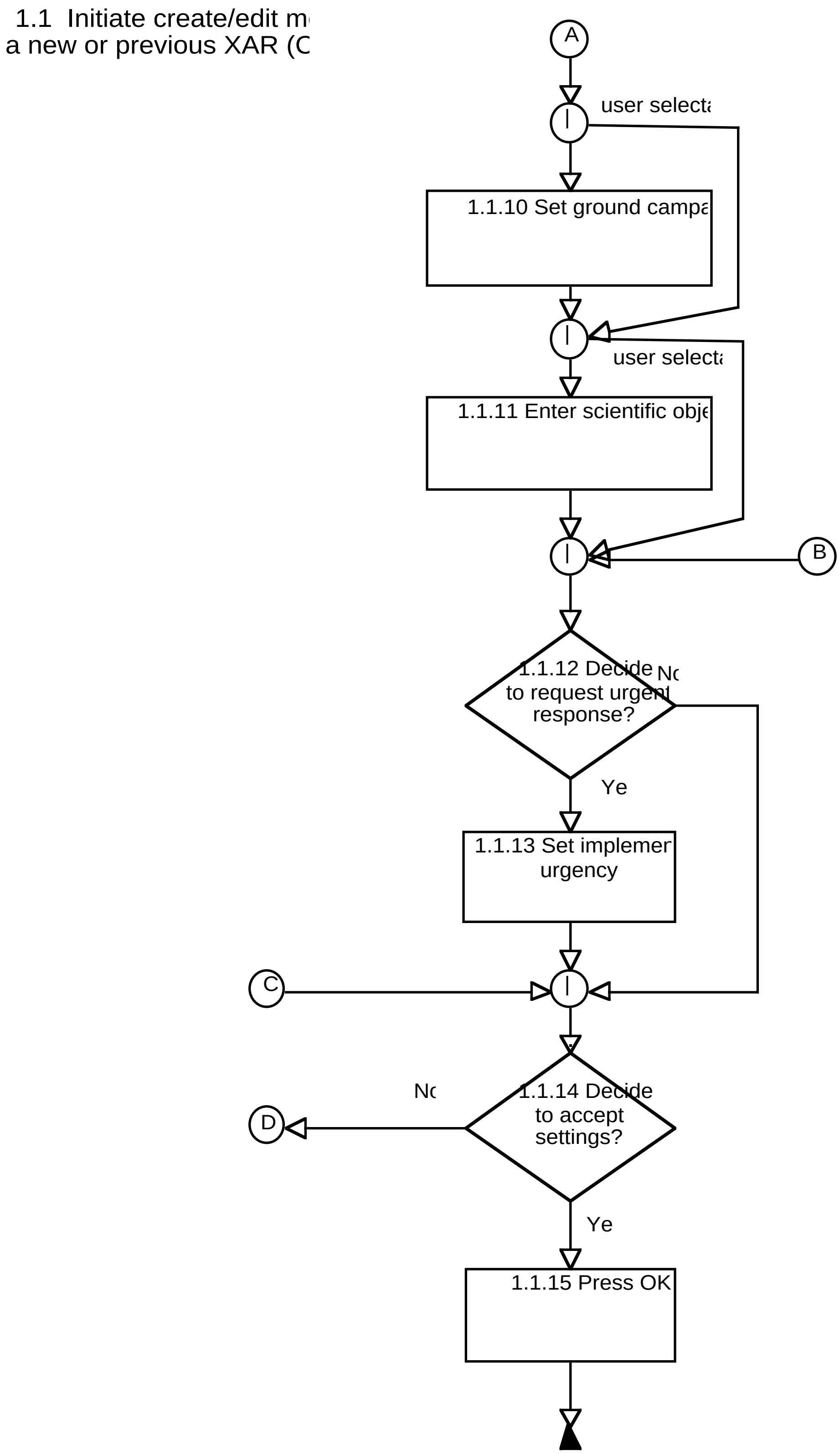


Figure 4-2. Detailed Workflow for DAR Function 1.1 (2 of 2)

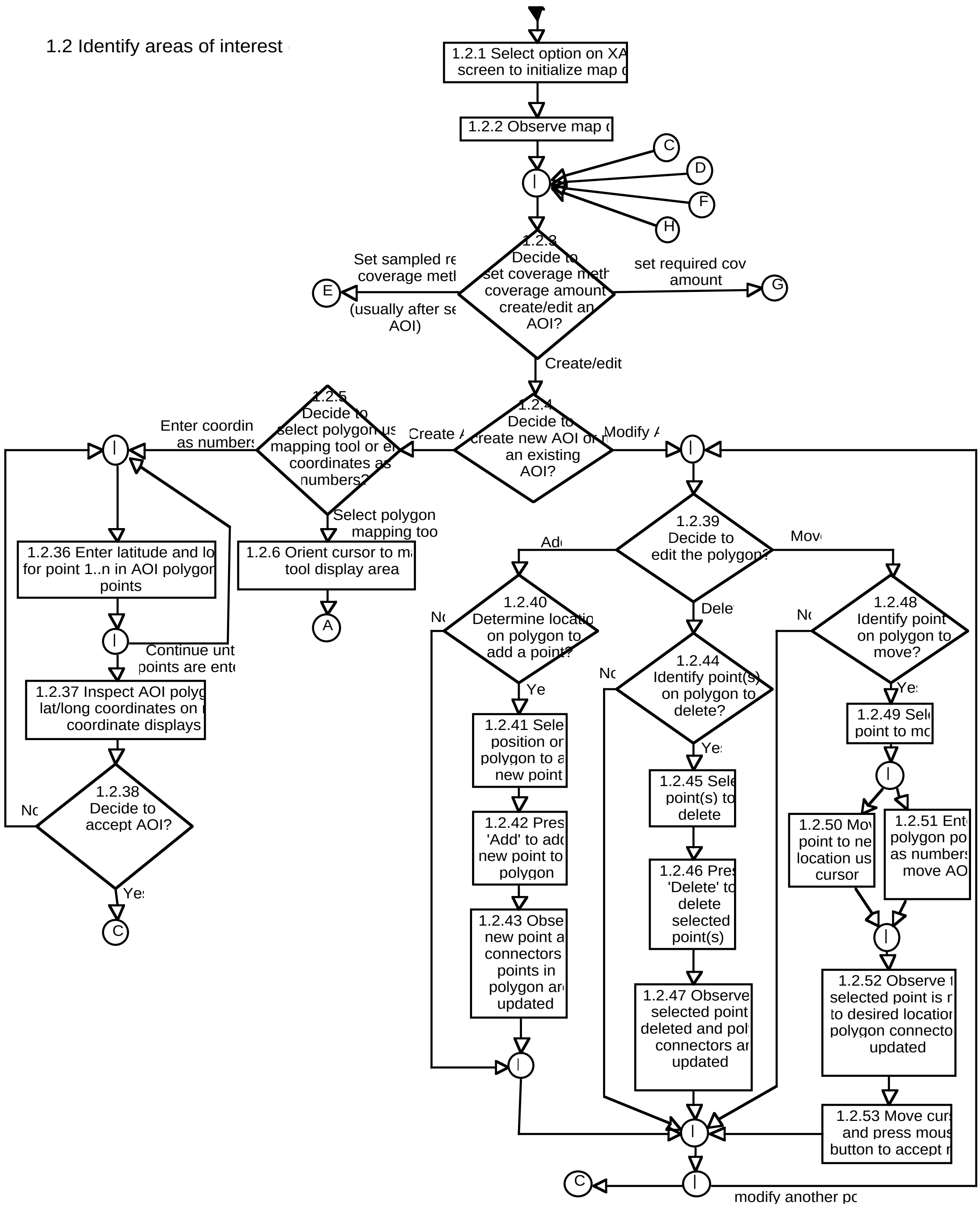
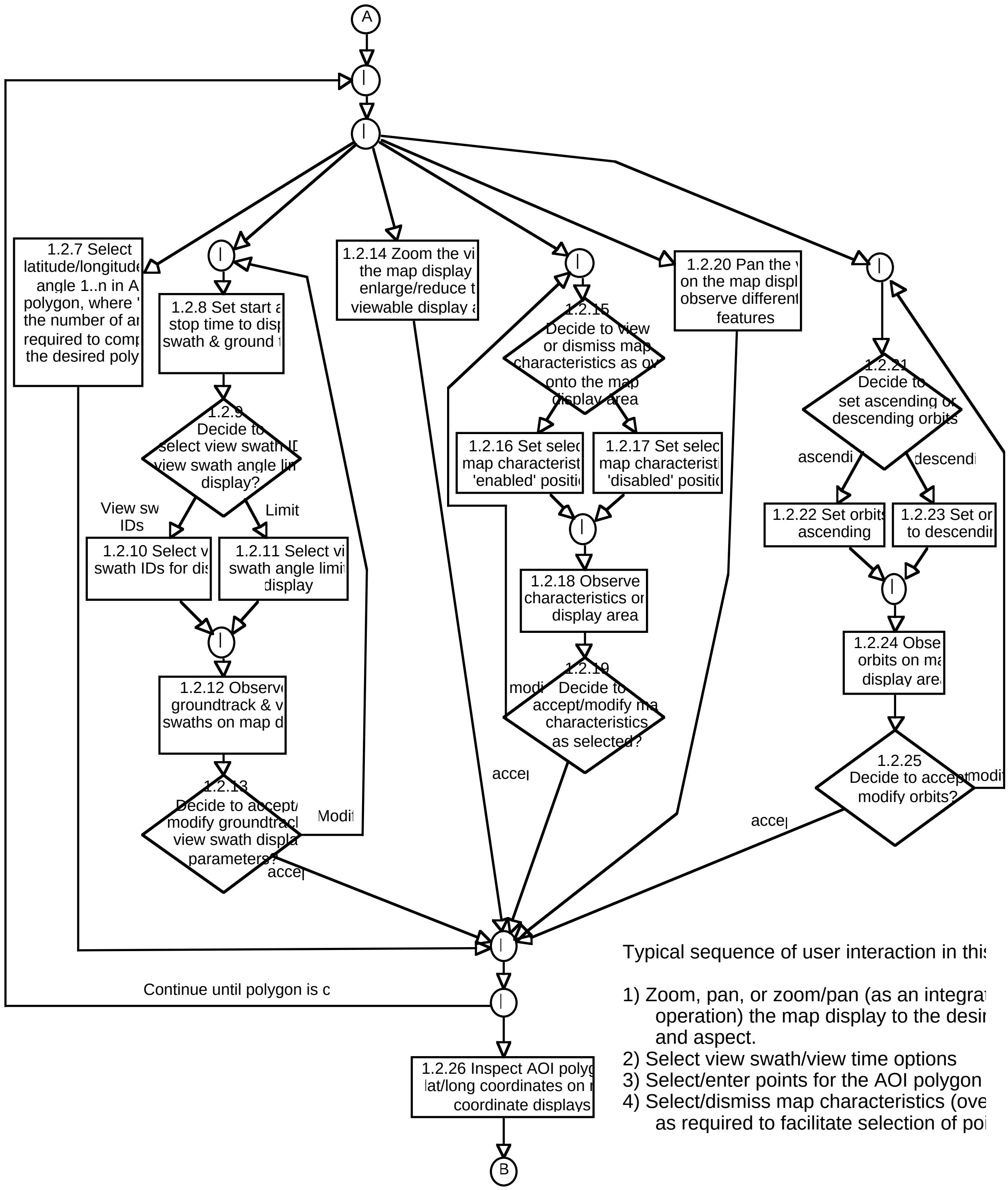


Figure 4-3. Detailed Workflow for DAR Function 1.2 (1 of 3)

1.2 Identify/modify area of interest



Typical sequence of user interaction in this

1) Zoom, pan, or zoom/pan (as an integra operation) the map display to the desir and aspect.
2) Select view swath/view time options
3) Select/enter points for the AOI polygon
4) Select/dismiss map characteristics (ove as required to facilitate selection of poi

Figure 4-3. Detailed Workflow for DAR Function 1.2. (2 of 3)

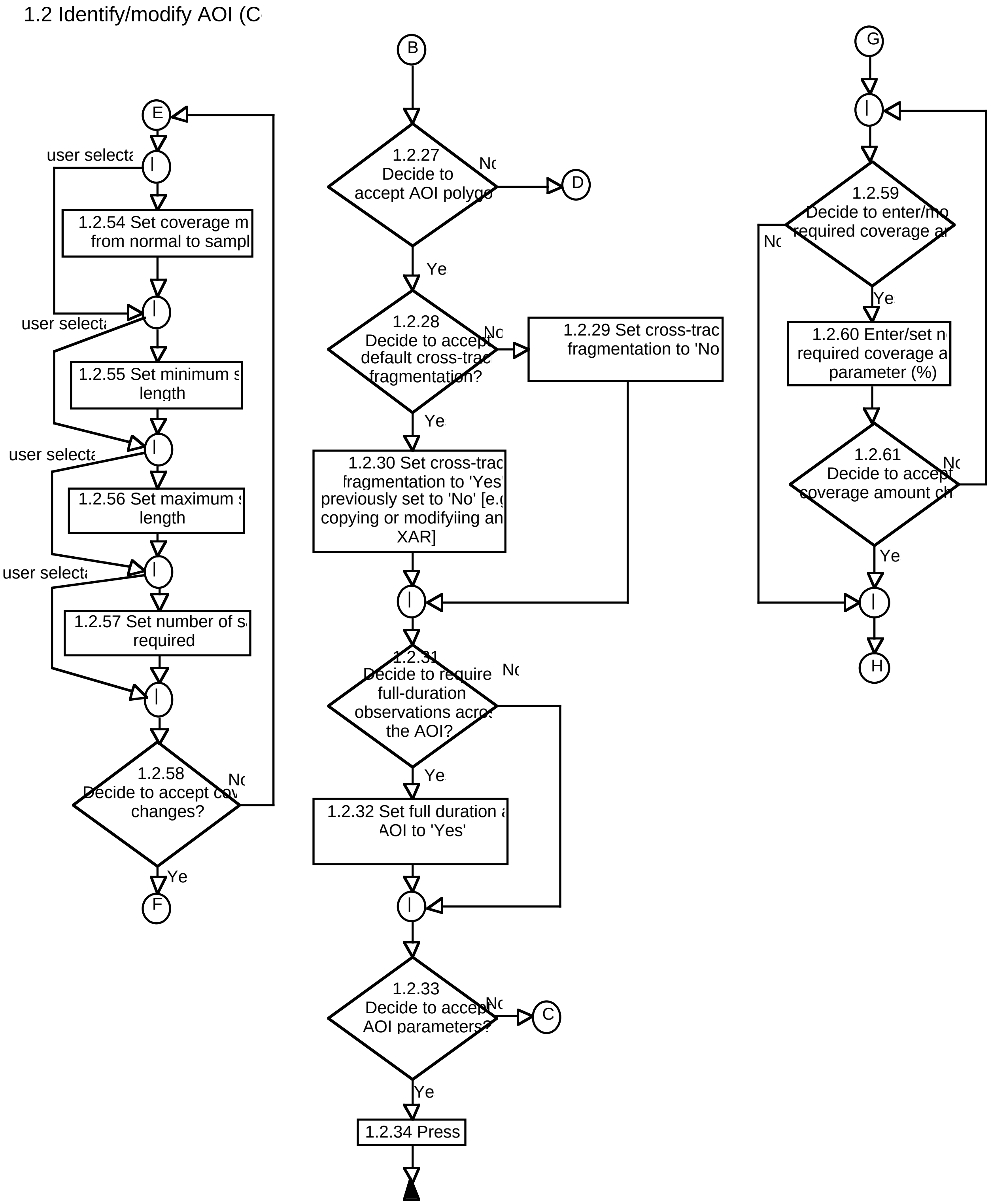


Figure 4-3. Detailed Workflow for DAR Function 1.2. (3 of 3)

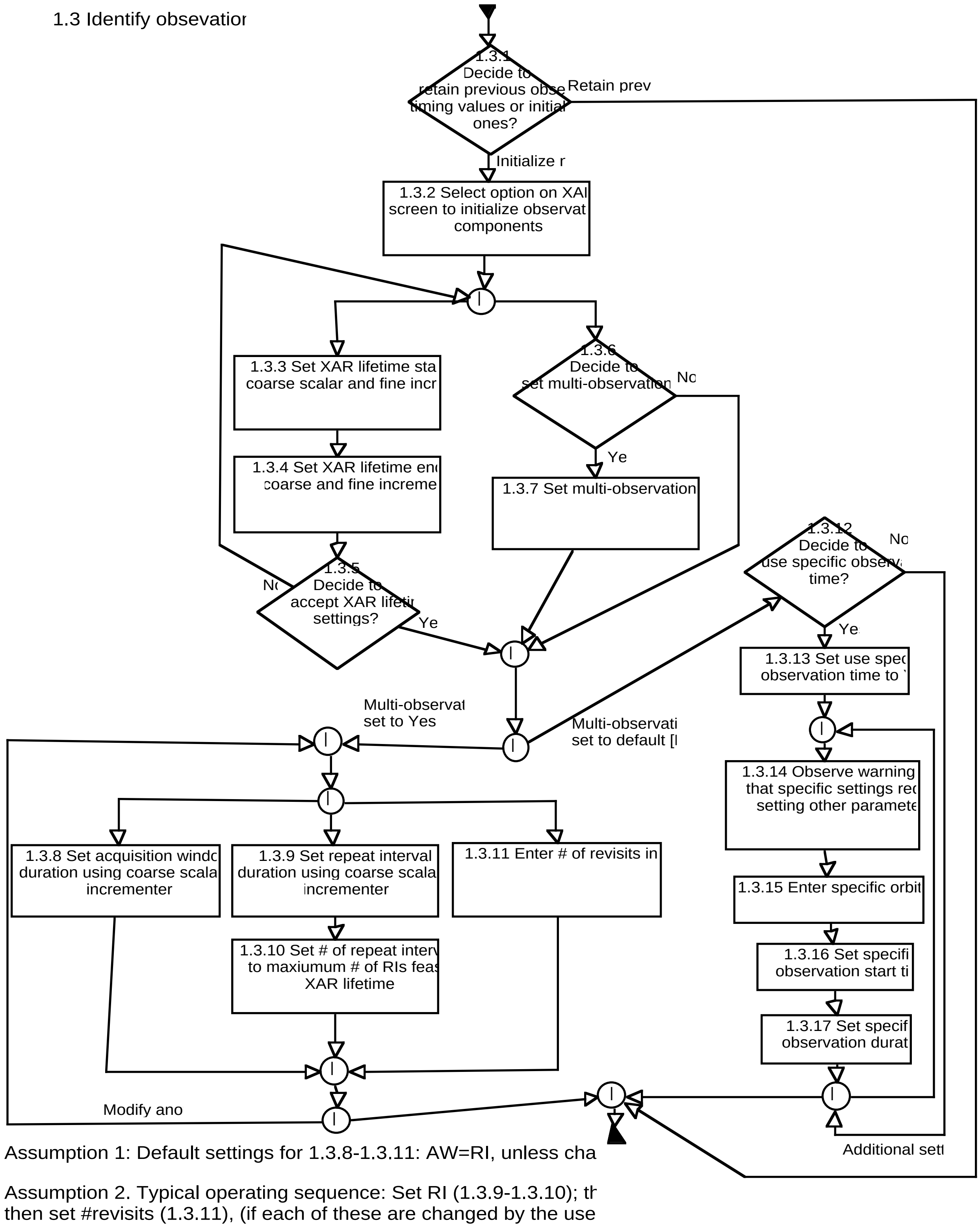


Figure 4-4. Detailed Workflow for DAR Function 1.3

1.4 Identify instrument con

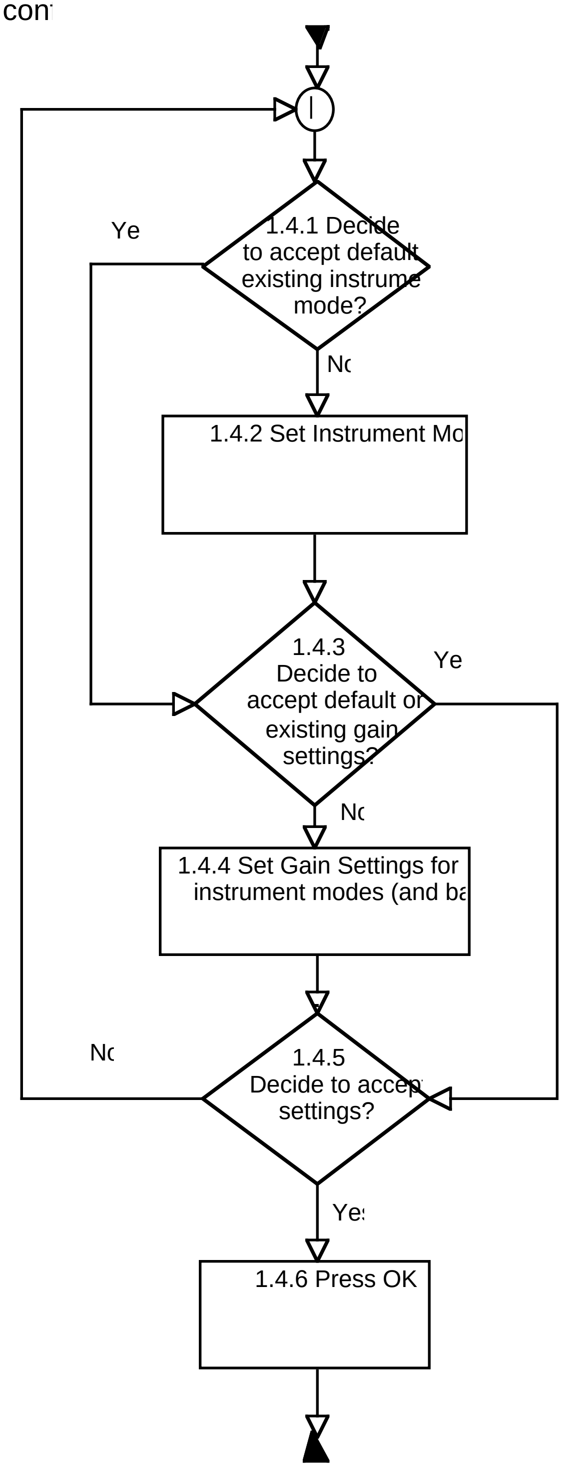


Figure 4-5. Detailed Workflow for DAR Function 1.4

1.5 Identify viewing ge

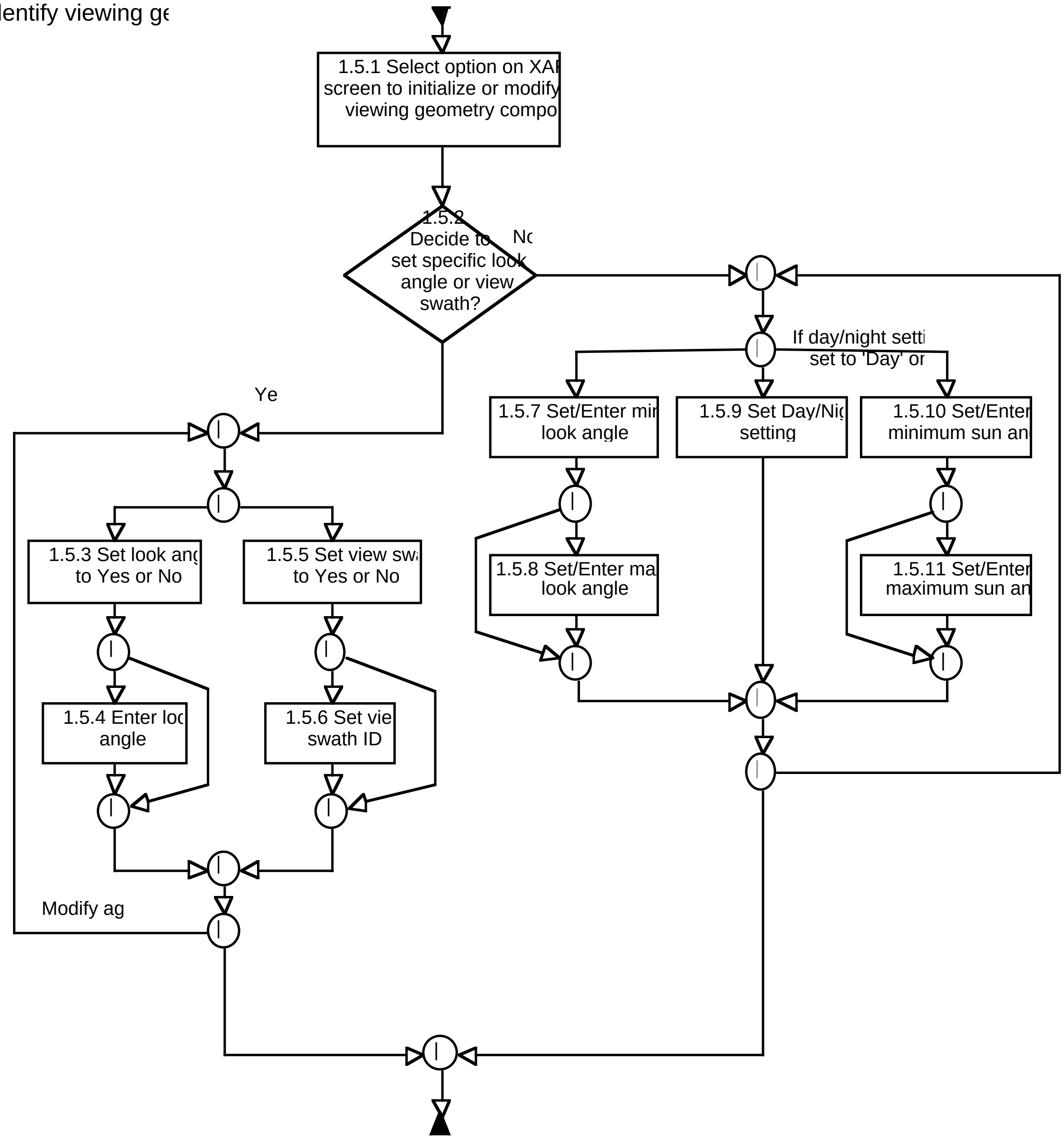


Figure 4-6. Detailed Workflow for DAR Function 1.5

1.6 Identify data c

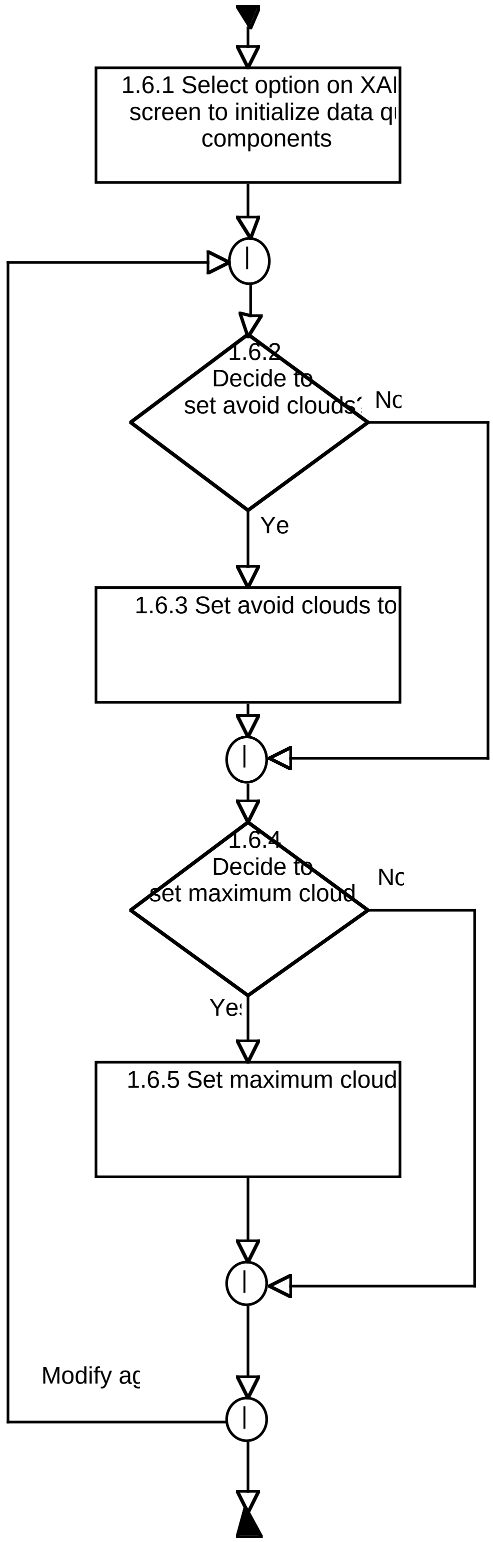


Figure 4-7. Detailed Workflow for DAR Function 1.6

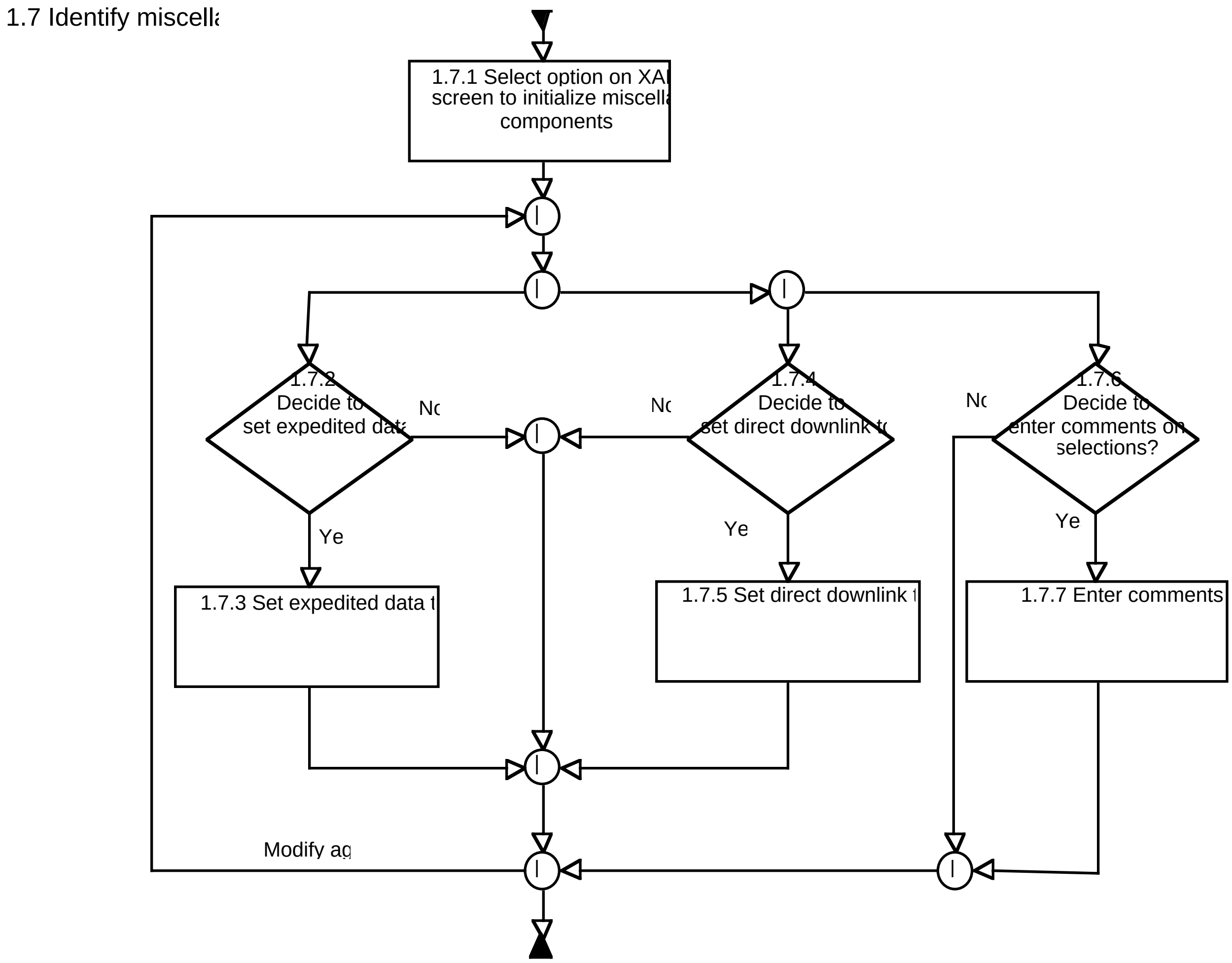


Figure 4-8. Detailed Workflow for DAR Function 1.7

1.8 Create/submit a DPR w

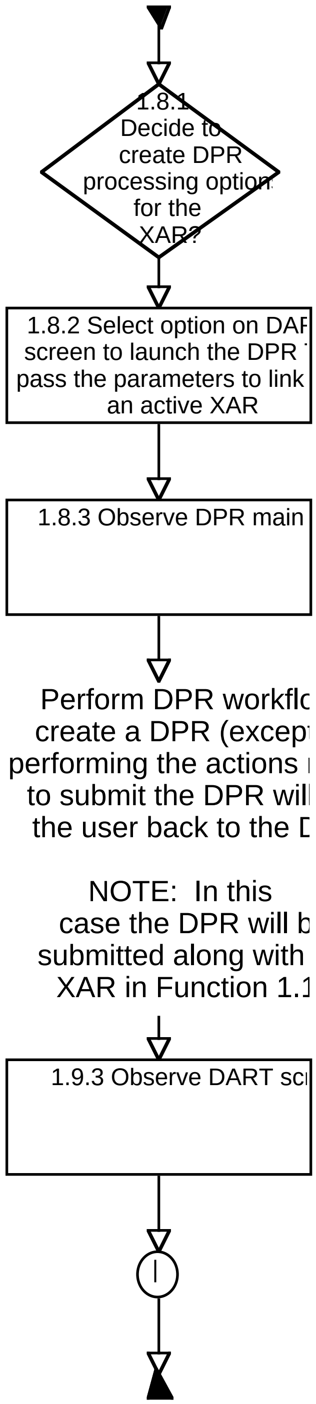


Figure 4-9. Detailed Workflow for DAR Function 1.8

1.9 Save local cop
[of work in progress
of XAR configuration pa

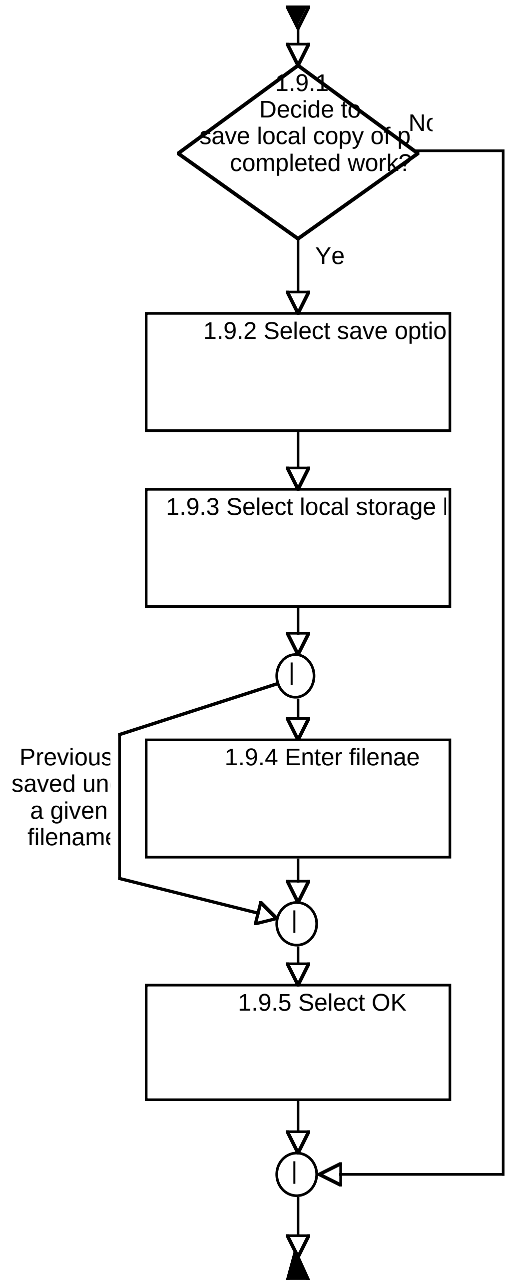


Figure 4-10. Detailed Workflow for DAR Function 1.9

1.10 Estimate resources required
ASTER product in accordance

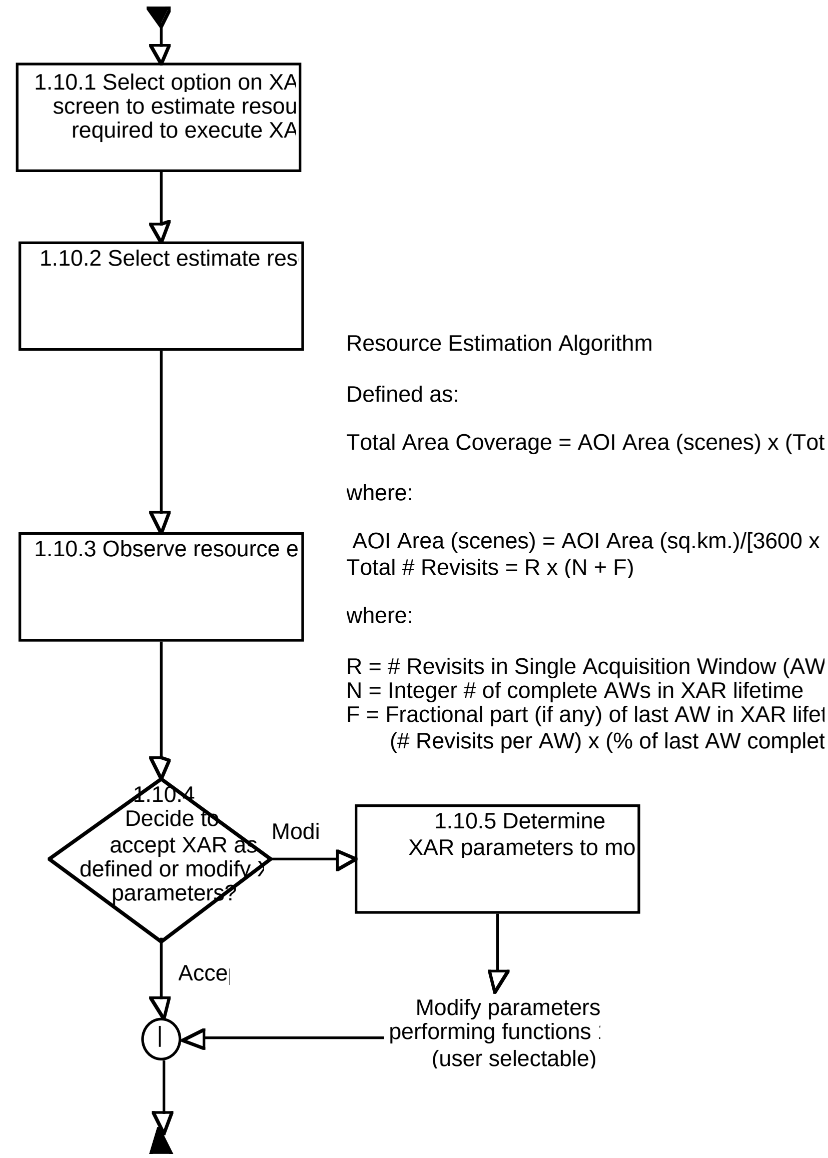


Figure 4-11. Detailed Workflow for DAR Function 1.10

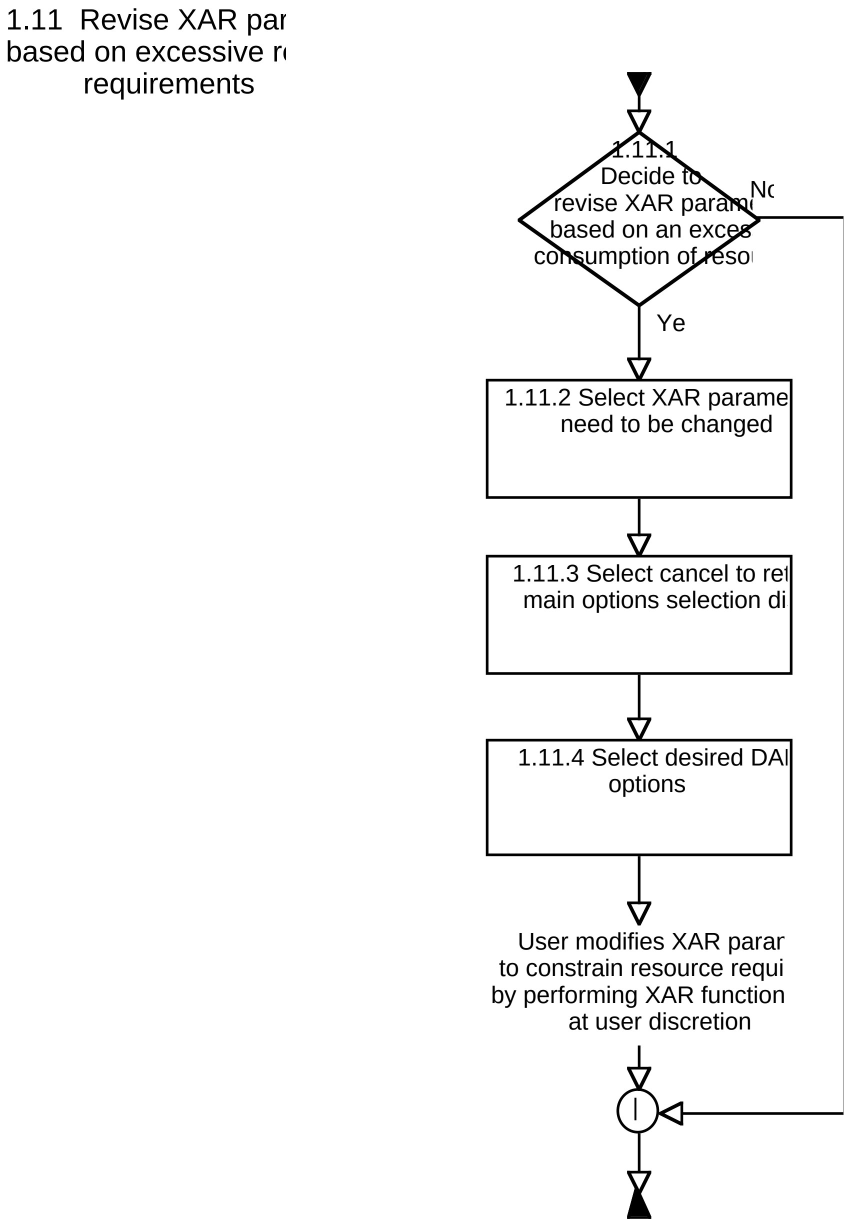


Figure 4-12. Detailed Workflow for DAR Function 1.11

1.12 Submit a

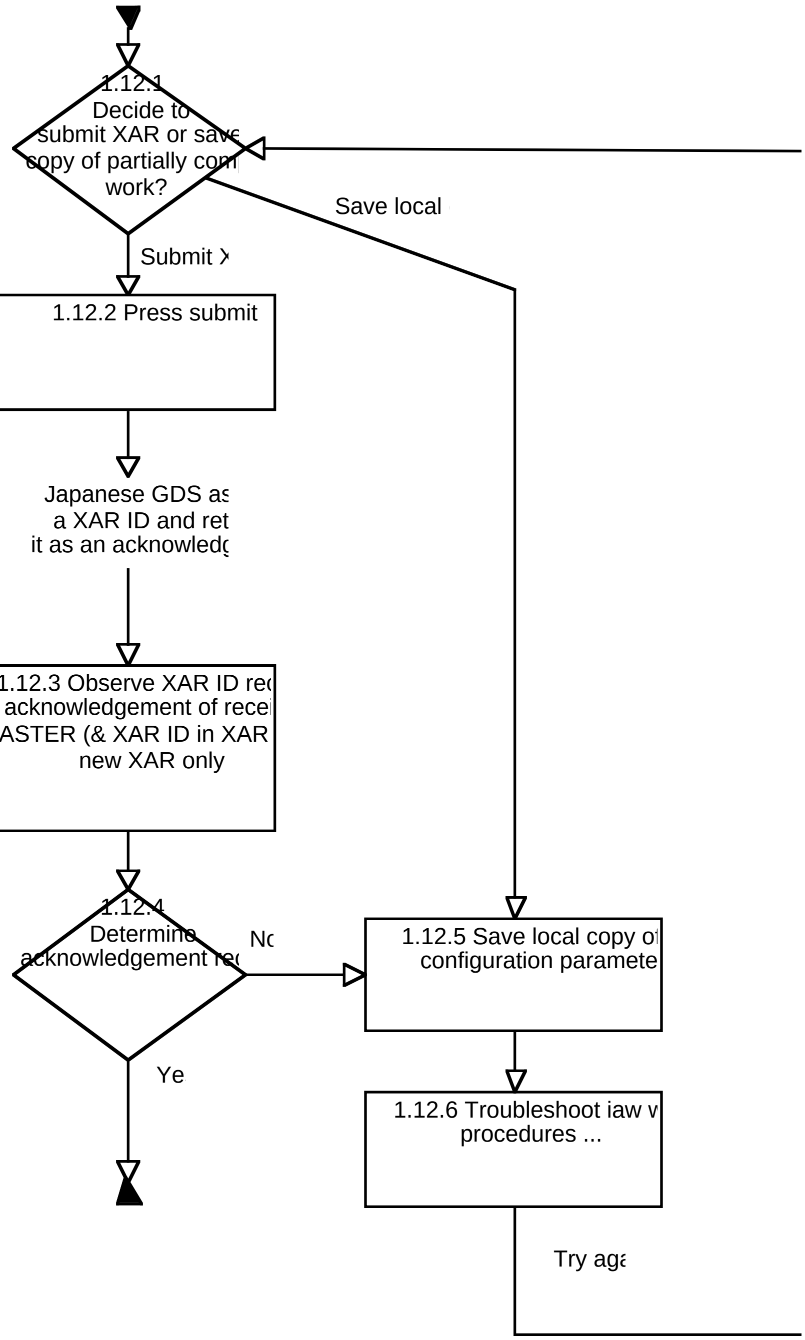


Figure 4-13. Detailed Workflows for DAR Function 1.12

4.1.3.2 Workflow Diagrams for DAR Function 2.0: Create/Submit a XAR Query

Figure 4-14 presents the top-level workflow diagram for DAR Function 2.0. Each of the remaining figures (4-15 through 4-19) present the workflows necessary for end-users to execute the DAR Functions 2.1 through 2.13 using the Release B XAR client query tool. Some of the functions depicted in figure 4-14 have no detailed workflow diagrams associated with them. This occurs in situations where the top-level workflow adequately covers the detailed DAR function without further elaboration or where the action defined by the workflow involve purely user actions (i.e., functions that identify human decision making).

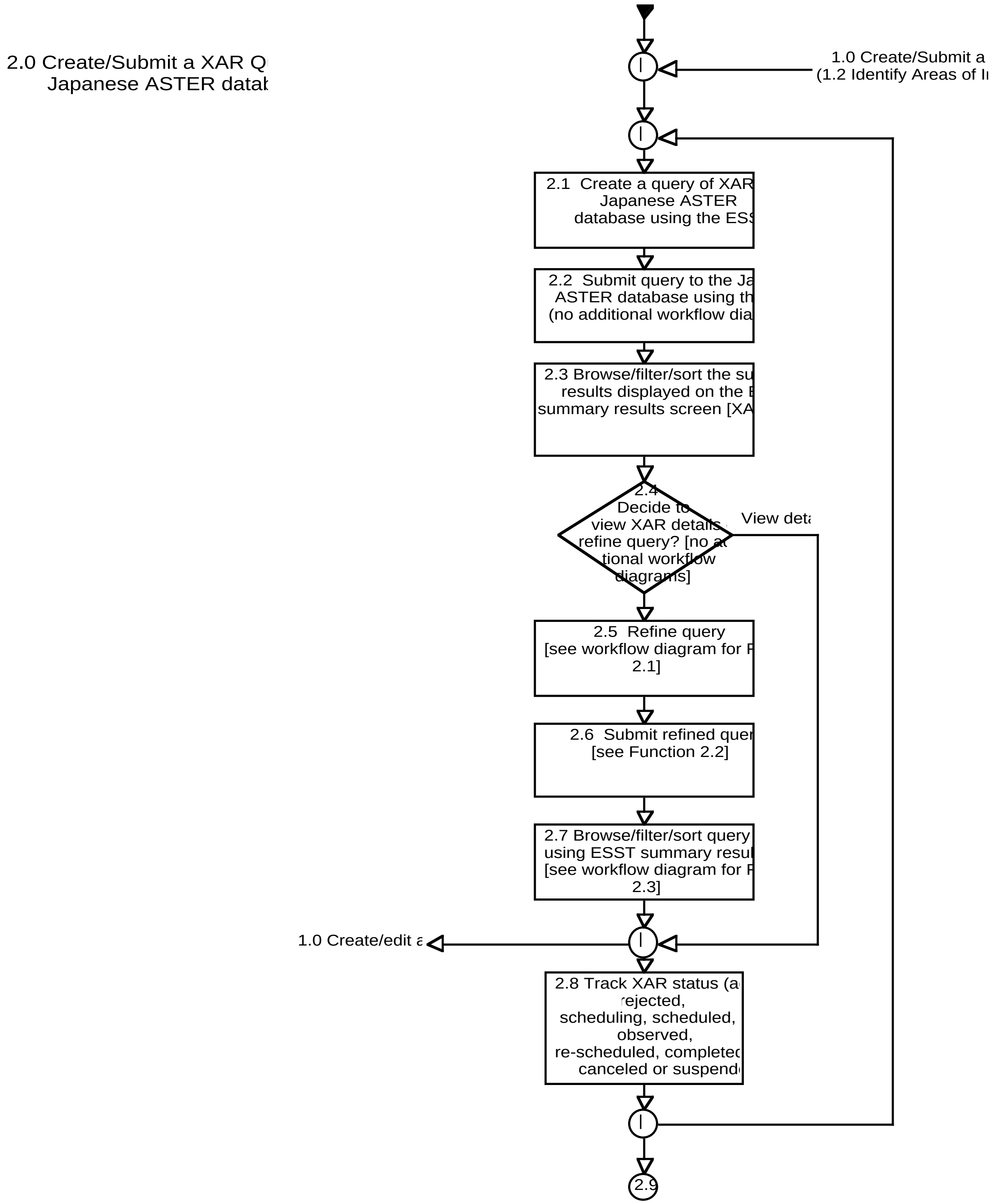


Figure 4-14. Top-level Workflow for DAR Function 2.0 (1 of 2)

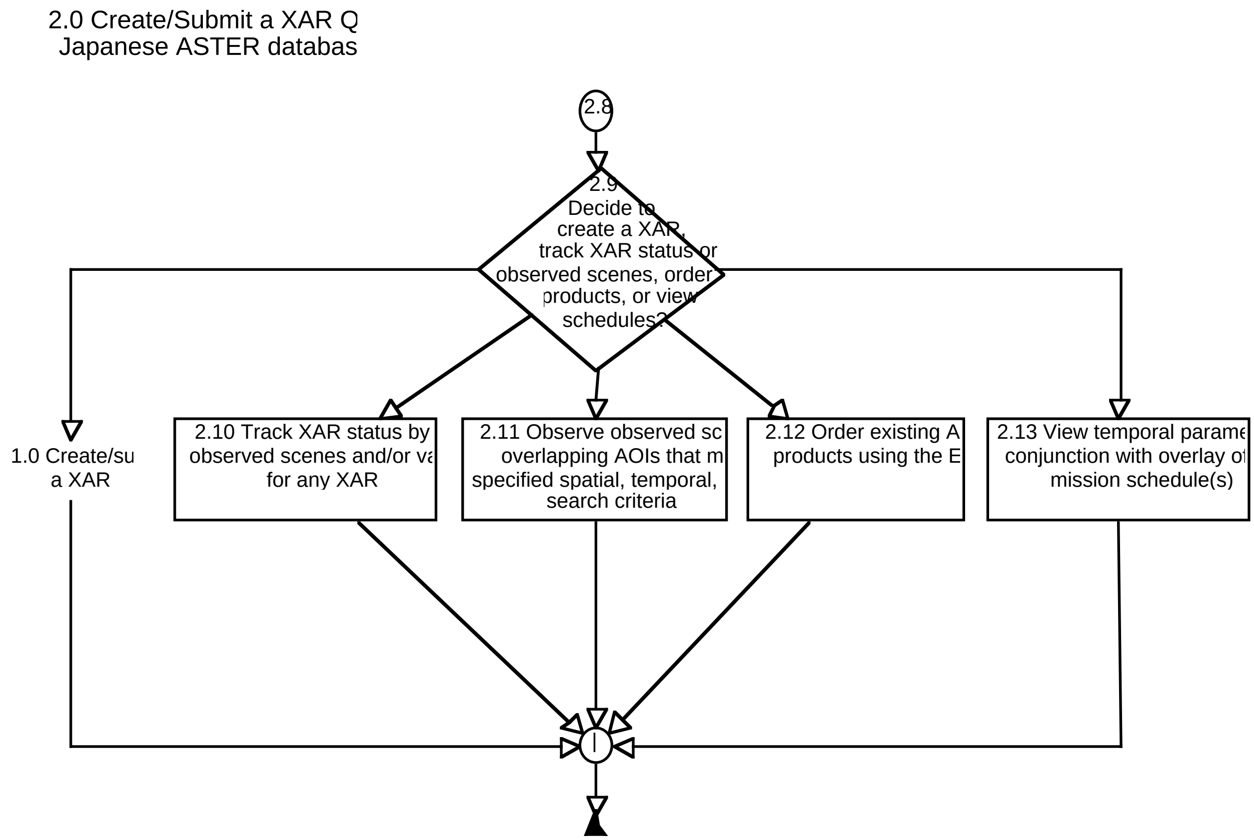


Figure 4-14. Top-level Workflow for DAR Function 2.0 (2 of 2)

The following seven figures present the detailed workflow diagrams for DAR Function 2.0. In the case of DAR Function 2.1 (see figure 4-15 below), three detailed workflow diagrams were defined for DAR Subfunctions 2.1.4, 2.1.5, and 2.1.6. This was necessary to reflect the peculiar levels of complexity associated with this function and these specific subfunctions. The workflow diagrams for these subfunctions are presented in figures 4-15a, b, and c.

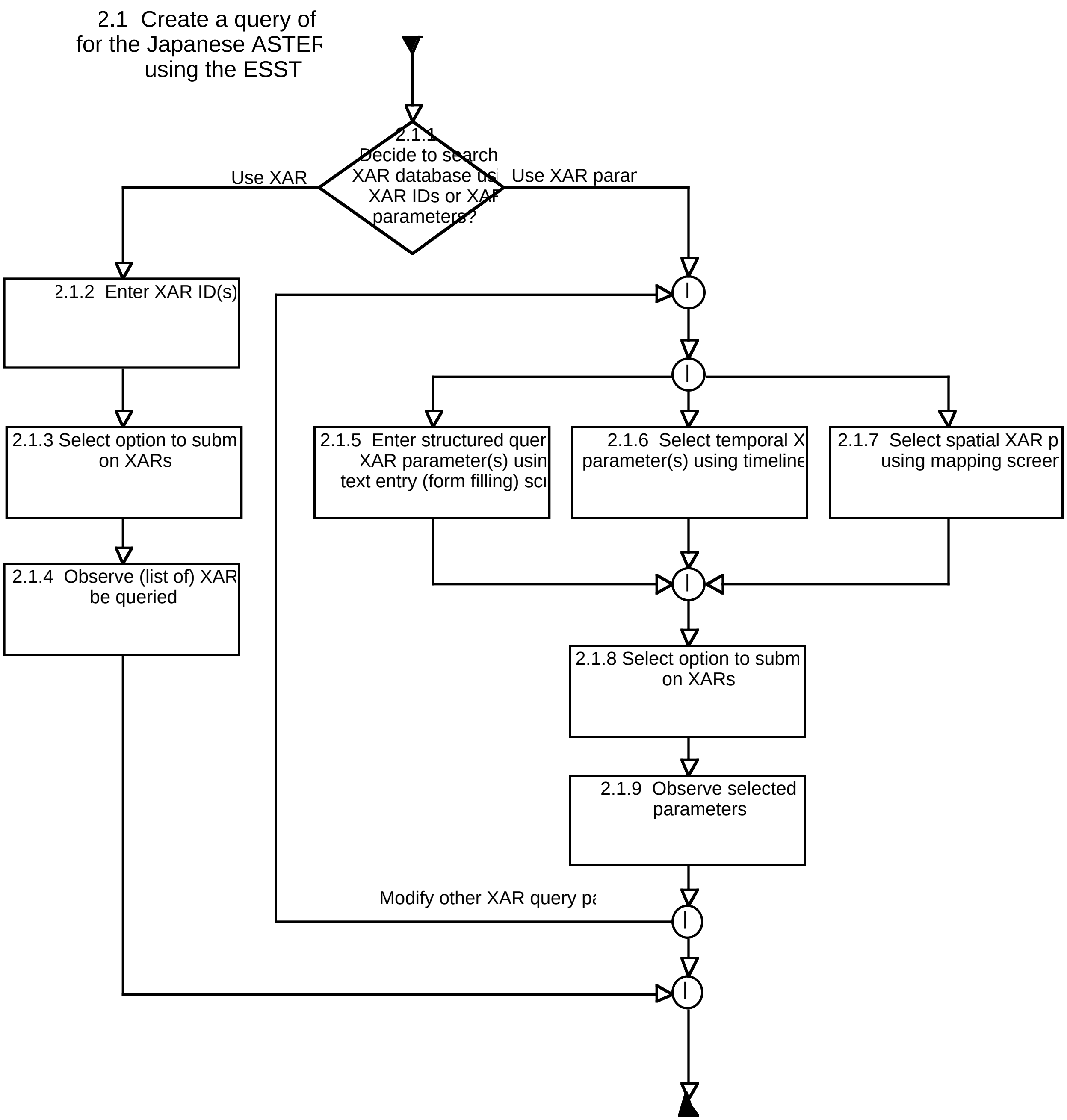


Figure 4-15. Detailed Workflow for DAR Function 2.1

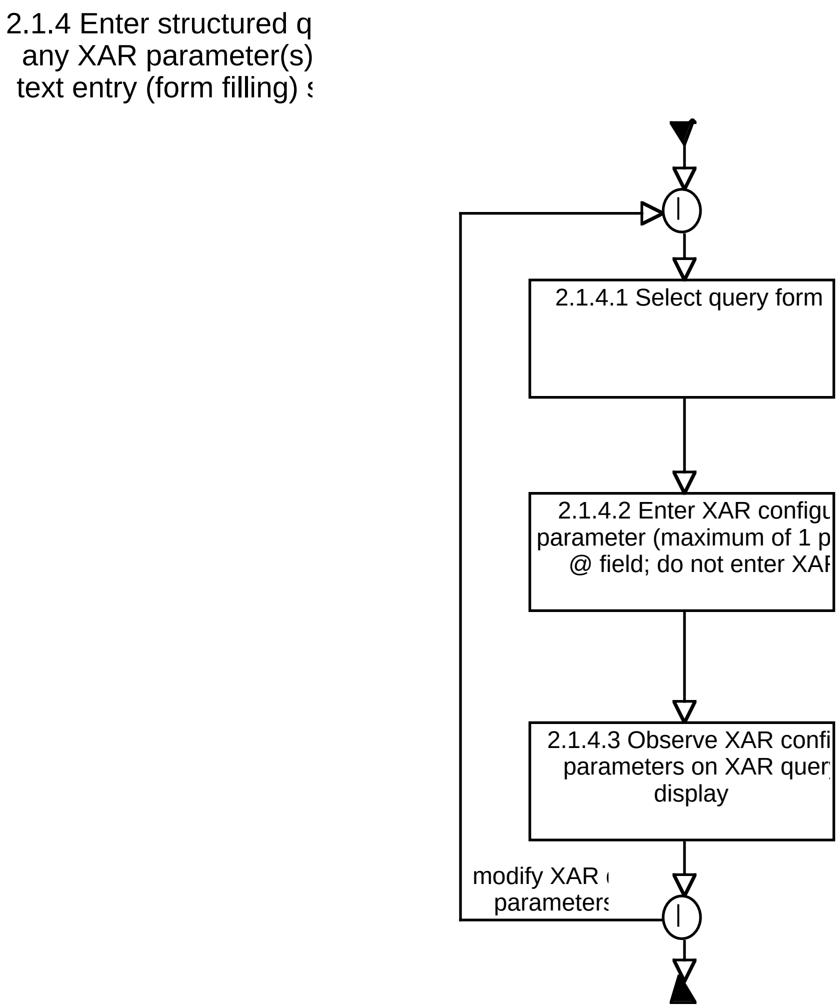


Figure 4-15a. Detailed Workflow for DAR Subfunction 2.1.4

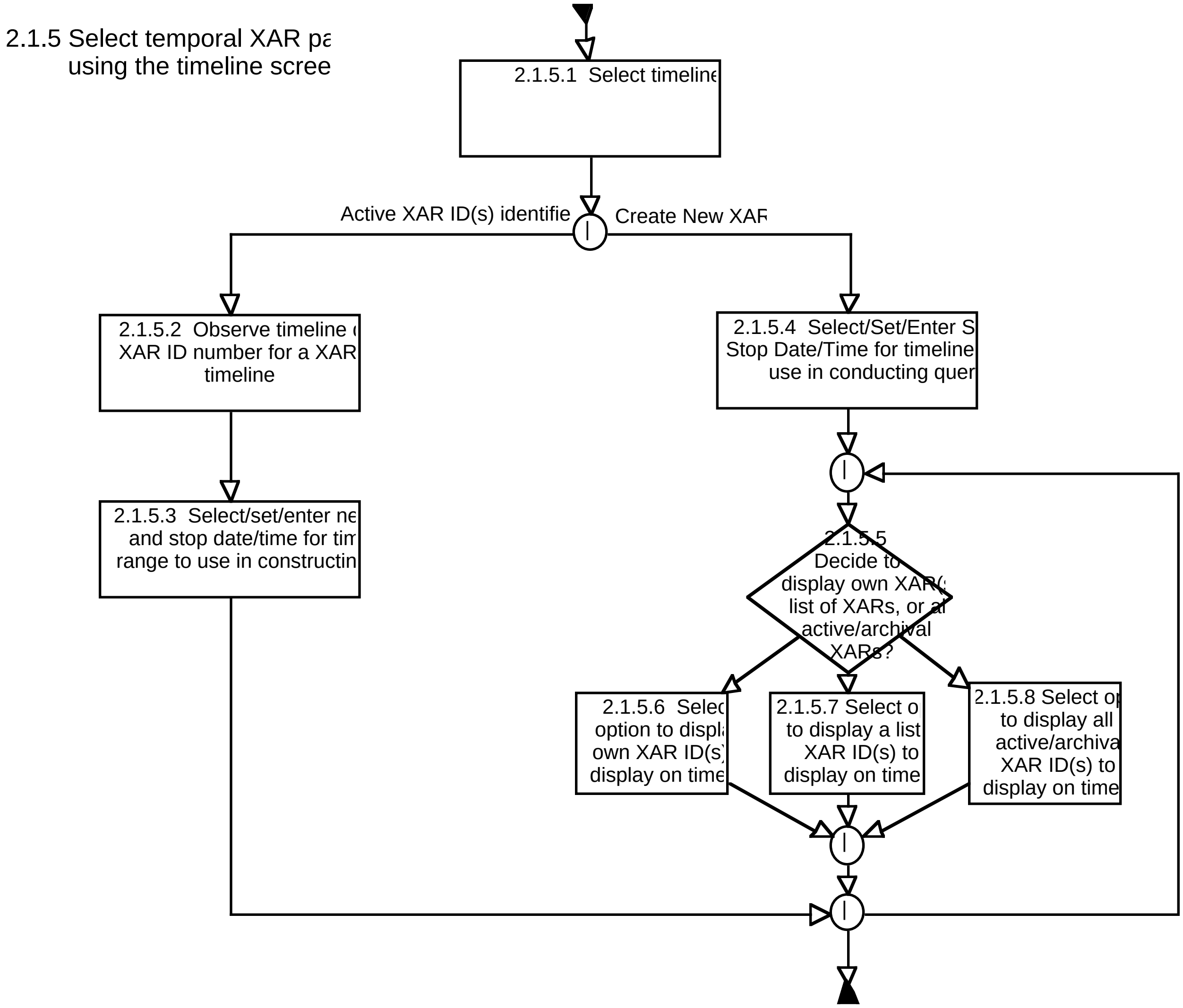


Figure 4-15b. Detailed Workflow for DAR Subfunction 2.1.5

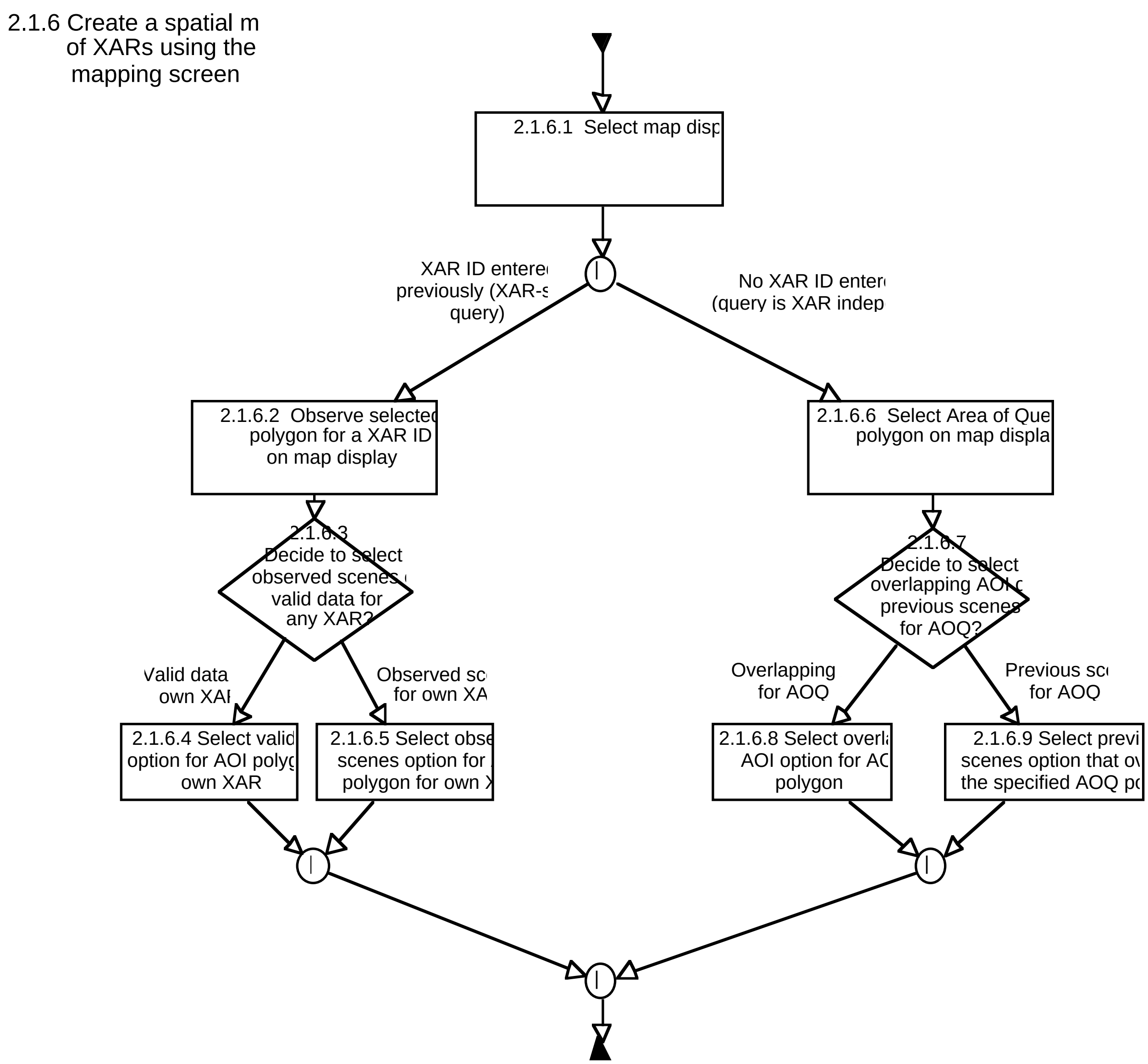


Figure 4-15c. Detailed Workflow for DAR Subfunction 2.1.6

2.3 Browse/filter/sort the summar
displayed on the ESST summary r
[XAR by XAR]

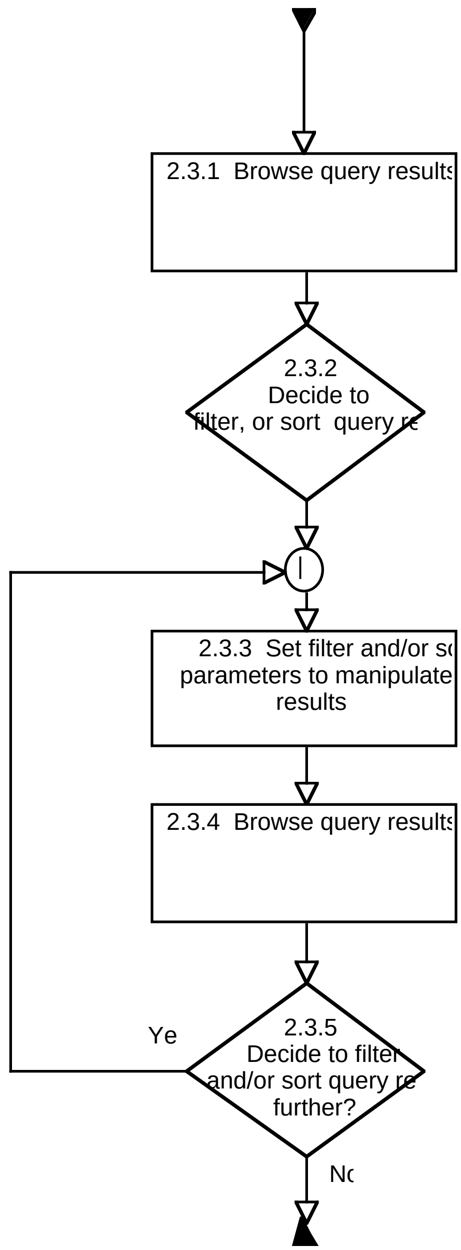


Figure 4-16. Detailed Workflow for DAR Function 2.3

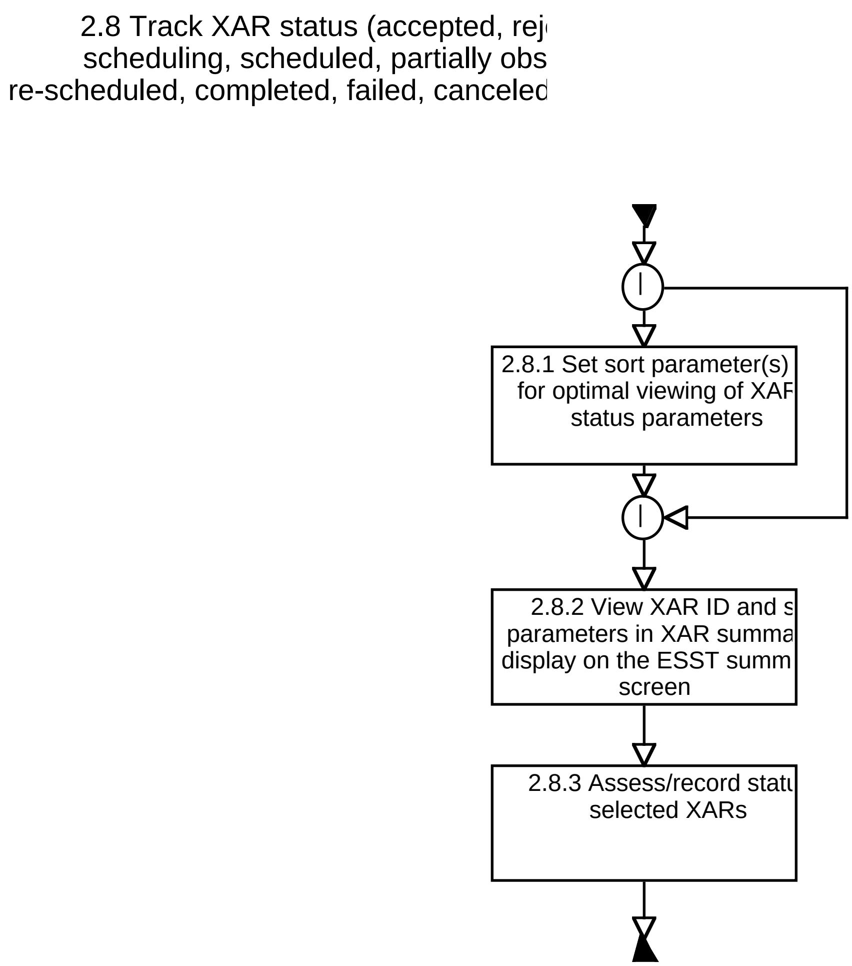


Figure 4-17. Detailed Workflow for DAR Function 2.8

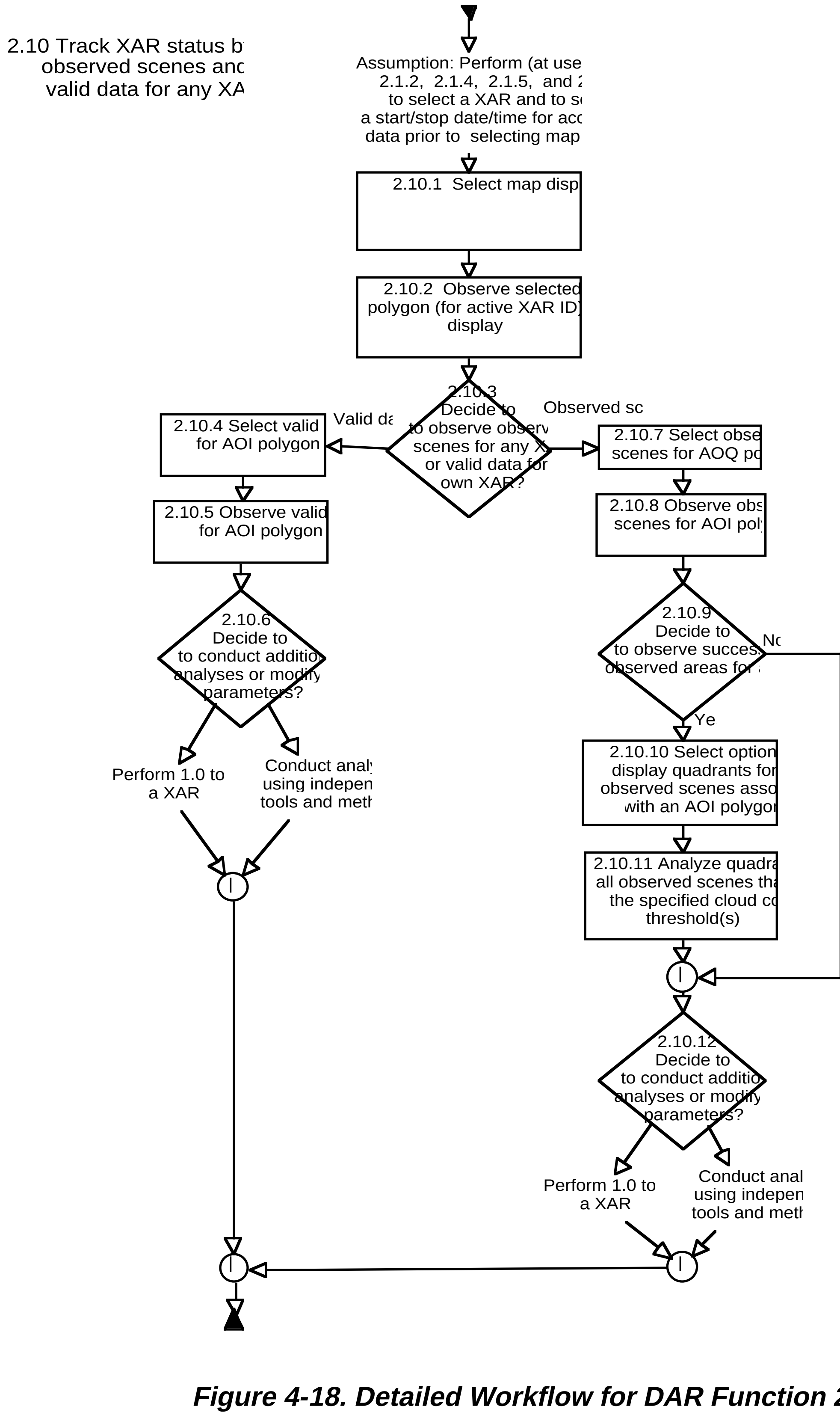


Figure 4-18. Detailed Workflow for DAR Function 2.10

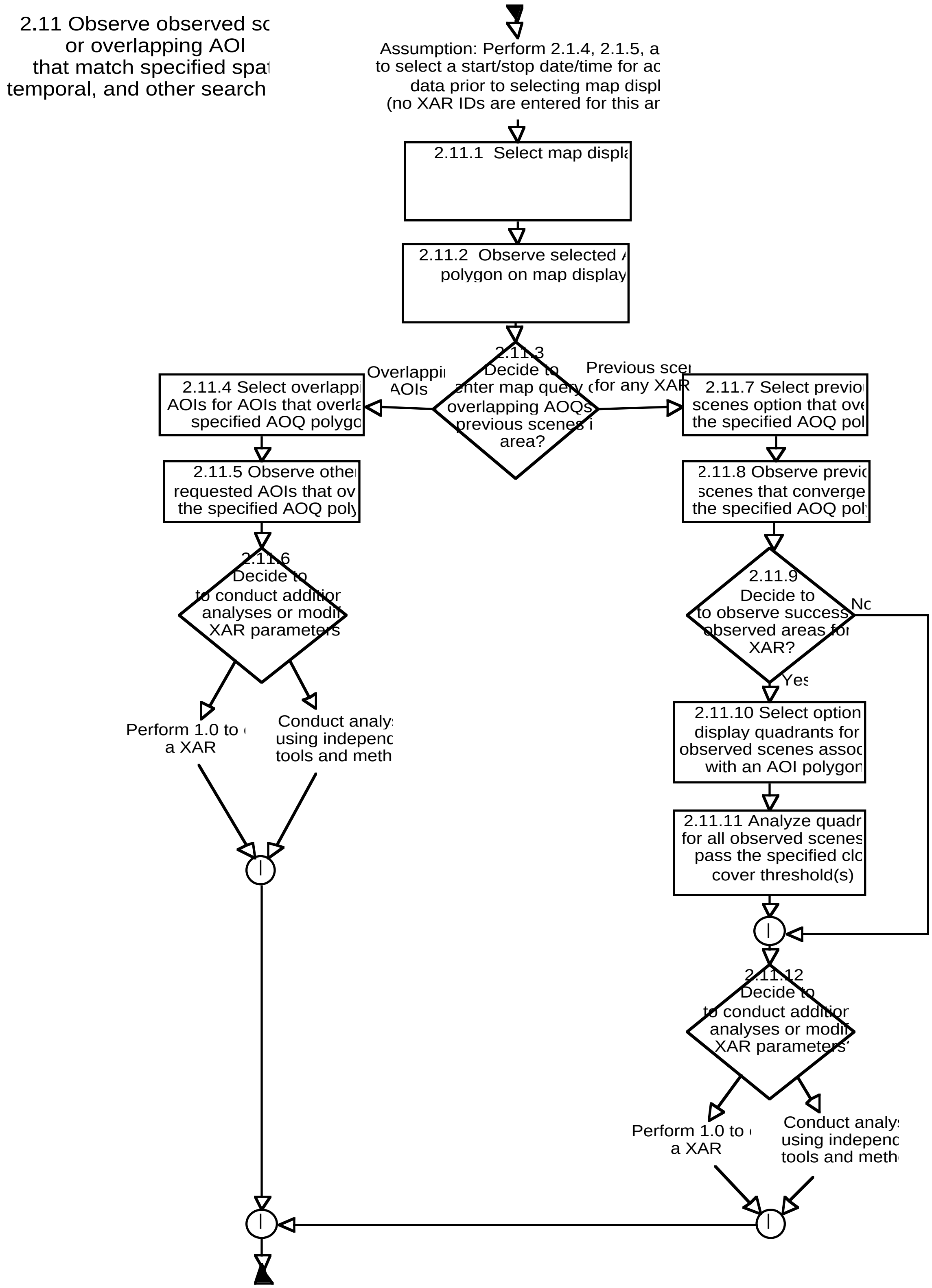


Figure 4-19. Detailed Workflow for DAR Function 2.11

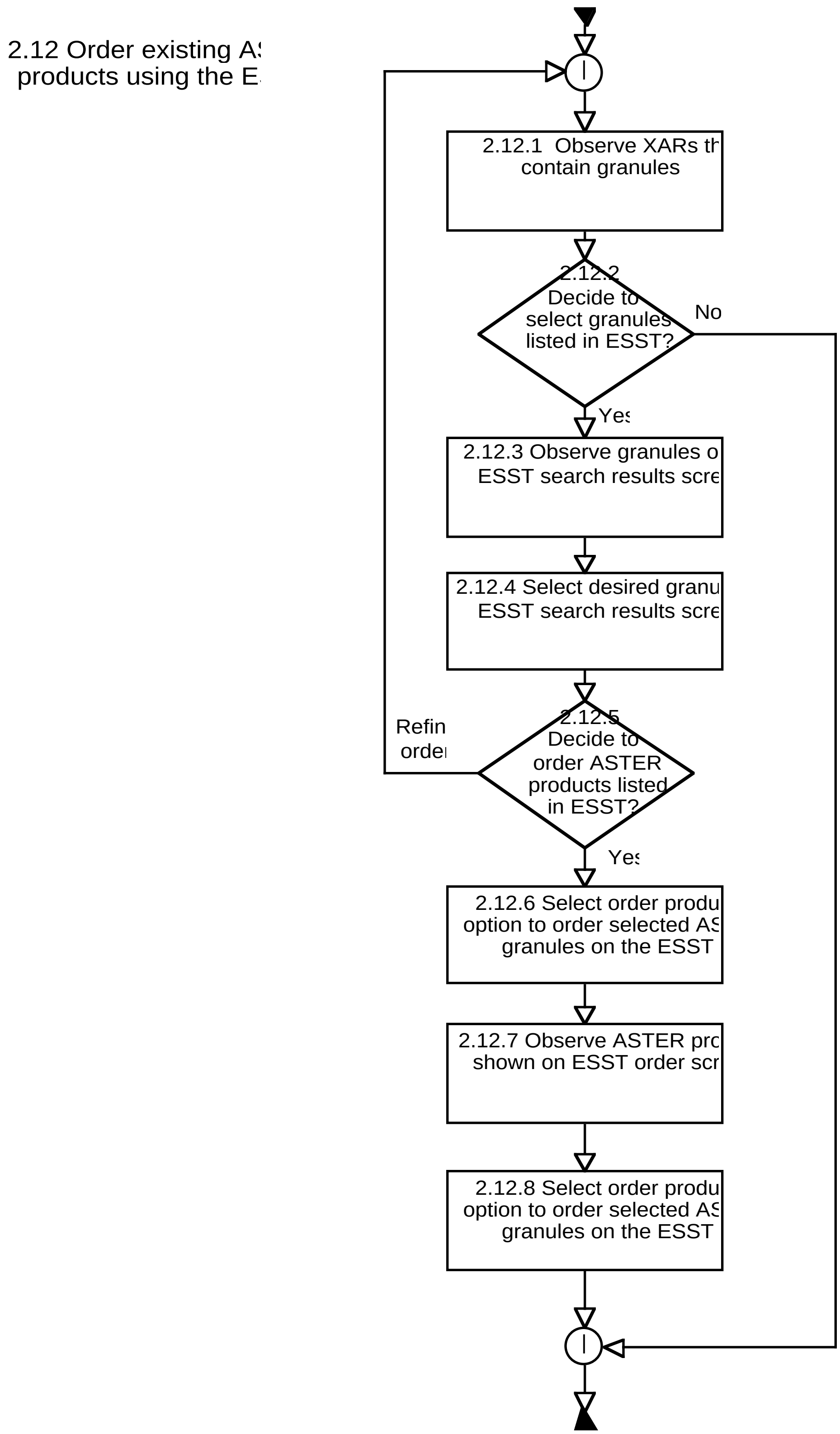


Figure 4-20. Detailed Workflow for DAR Function 2.12

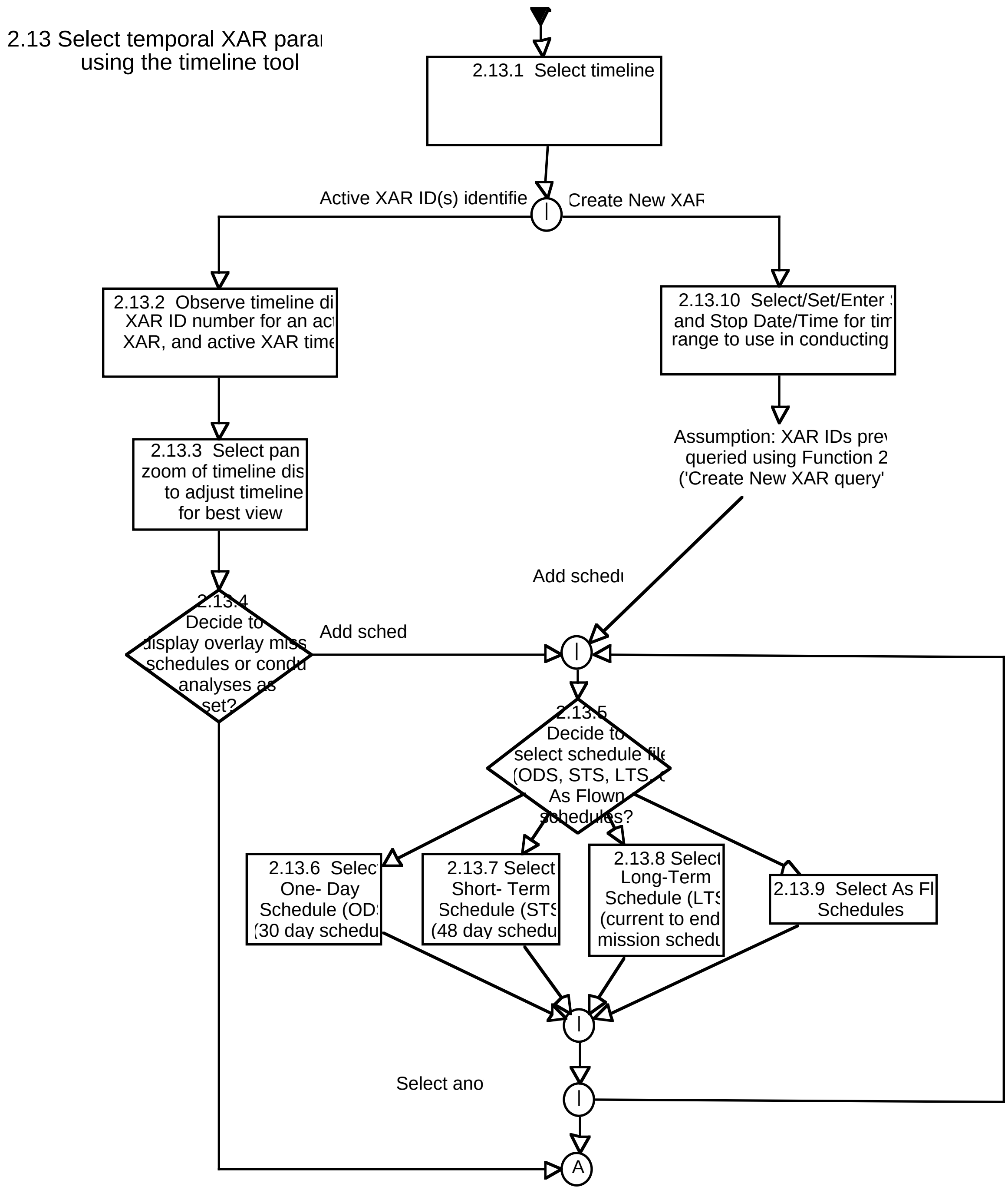


Figure 4-21. Detailed Workflow for DAR Function 2.13 (1 of 2)

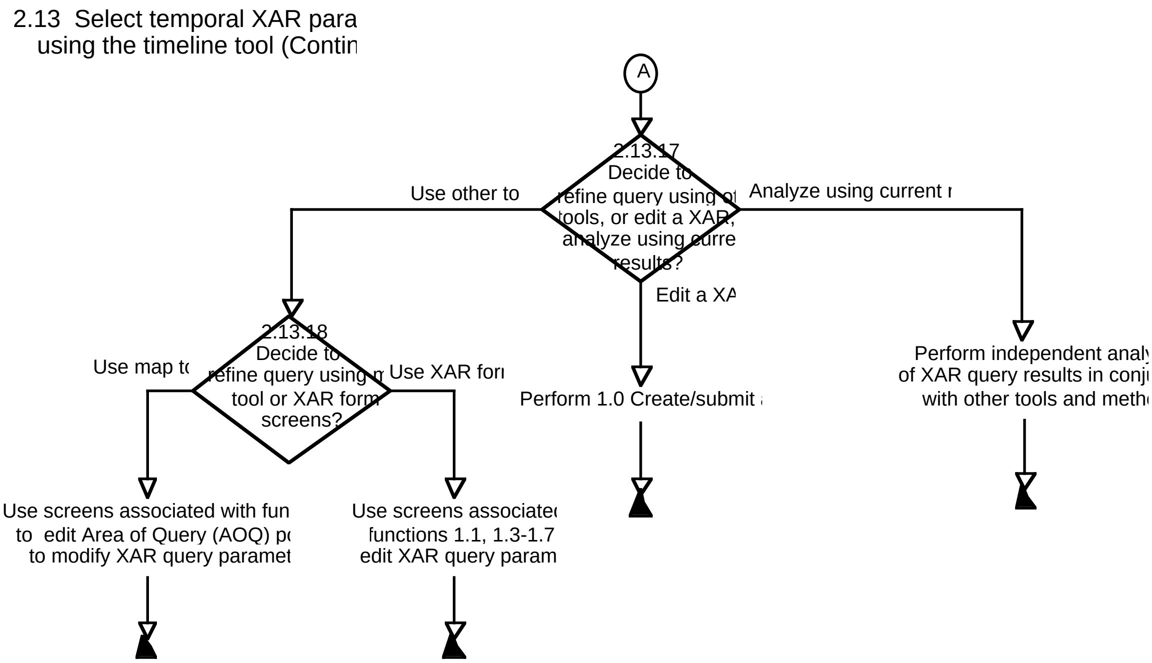


Figure 4-21. Detailed Workflow for DAR Function 2.13 (2 of 2)

4.2 Timeline Software

A portion of the prototyping effort was devoted to developing code to implement the essential requirements of a timeline. After analysis of several alternative packages, it was found that the timeline capabilities provided in Delphi code libraries fully provide the functionality needed by the DAR tool. A prototype of the timeline was coded, and it was loosely linked into the DAR GUI prototype. The prototyped timeline as coded has the following capabilities:

- Display the instrument and AM-1 spacecraft activities and related information chronologically on a schedule timeline,
- Display any user-defined combination of the activities and useful messages such as equator crossings, terminator crossings and mode changes etc.,
- Display textual information on any activities,
- Display time in GMT, local solar time or orbit-relative times,
- Display the time interval of an event and the time interval between the neighboring events,
- Filter observation data displayed on a schedule timeline,
- Allow the user to control the scale of the timeline,

- Allow the user to continuously scroll through the timeline in two directions,
- Show the user which events take place at a given time instant.

Since the timeline requirements for the DAR were more demanding than those of the general ESST, it is expected that the Delphi libraries will be able to fully meet ESST timeline requirements. This allows one software library to support all Client timeline functionality.

4.3 As-built Object Model

An object model was developed for the DAR code using the reverse-engineering capabilites of the Rational Rose Object Modeling Tool. The object model was derived from a earlier version of the DAR tool, and has not been updated for the current version. Specific areas in which the object model needs updating are spatial capabilities, and the back-end database, which was implemented for the prototype but will not be used in the production version.

Given the reverse engineering capabilities of the Rational Tool, it was not felt necessary to include the object model in this document. Note of it is made here to make future developers aware of the fact that an object models exists, and although it is not up-to-date it would serve as a baseline for further object modeling work. Details on accessing the object model can be obtained by contacting the author of this paper.

This page intentionally left blank.

5. Development Notes

5.1 Development Areas

The DAR prototype did resolve all design and development issues for the DAR tool. Further work should be undertaken in the following areas.

5.1.1 Human Factors Engineering (HFE) Considerations and GUI Consistency with the DAR Workflows

The workflow analysis, described in section 4., was undertaken with the expectation that the DAR workflows would provide a basis for streamlining the 'look-and-feel' of the DAR tool GUIs. This represents an additional area of development for the DAR Tool. The discussion that follows describes the similarities and differences between the workflow for DAR Function 1.0 and the current DAR Tool prototype that was developed during this period of performance. This information was extracted from a human factor engineering (HFE) assessment of the DAR Tool prototype GUIs.

5.1.1.1 HFE Assessment of the DAR Tool Prototype GUI Screens

This section presents the results of a HFE assessment of the DAR Tool prototype GUIs. This discussion begins with a list of the HFE comments that applies to most of the GUIs/dialogs contained in the current prototype. The rest of the discussion in this section is organized serially in accordance with each of the GUIs/dialogs contained in the prototype. Screen dump representations of each of these GUIs/dialogs are shown in Appendix A.[1] The assessment for each GUI/dialog is divided into two parts. The first part presents human engineering comments on the GUI/dialog. The second part addresses (in a table for each GUI/dialog) the degree to which the current set of DAR prototype GUIs/dialogs match or mismatch the associated workflow diagrams for DAR Function 1.0.

5.1.1.2 General Human Factors Engineering Comments

The following HFE comments are widely applicable to all DAR Tool prototype GUIs/dialogs. It is anticipated that these comments will be addressed, as appropriate, in a later effort to revise the DAR Tool prototype GUIs to implement the DAR workflows.

[1] Due to the iterative and somewhat incestuous nature of rapid prototyping, the GUIs/dialogs for the DAR Tool prototype actually shown in Appendix A are somewhat newer versions of the GUIs/dialogs than the GUIs/dialogs on which the human factors engineering (HFE) assessments were conducted. Where a GUI/dialog title in Appendix A is different between from that which was evaluated, such differences are noted in the text.

1. No visual cue is provided anywhere within this set of GUIs to let the user know what XAR they are working on; whether its new or previously existed; whether it was queried for or loaded locally.

2. GUI widgets should be aligned on all interfaces, in accordance with the formatting guidelines contained in the *ECS User Interface Style Guide*.

3. The proper case should be used for all interface labels.

4. All acronyms should be defined.

5. Use of option buttons in many places should be replaced with pick lists (the combo box widget). The Combo Box possesses the same basic functionality as the options button, but provides more room for developers to spell out acronyms that are currently undefined. Additionally, extensive use of pick lists is under consideration for the DPR tool and their consistent use would contribute to similar "look & feel."

6. Dialogs should not be colored sea blue, but rather Tan (RGB value #d1ac7f).

5.1.1.2.1 Main Window GUI and the "Identification & Tracking" Dialog

The 'Main Window GUI' is known as the '*Initial Screen*' in Appendix A. Likewise, the 'Identification and Tracking' dialog is known as the '*Identification and Classification screen*' in Appendix A.

a. Human factors engineering comments:

1. Assumption: creation of new XAR starts from the initial DART window.

2. Navigation is poorly implemented which leads to context confusion. Consequently, users would have difficulty:

   - Knowing where they currently are (i.e. What XAR they are manipulating?)

   - What action they are performing (there is no visual cue for XAR creation or editing)

   - Knowing how to initiate the creation of a XAR of the editing of existing XAR. There is no clear way either explained or visually prompted to execute these functions.

3. Discrete Attribute Summary, Spatial Summary, & Temporal Summary of what? If these are present on the Earth Science Search tool already, are they necessary here?

4. If there is more that one XAR whose attributes are summarized in the primary screen; which of these XARs is being displayed in the Spatial and Temporal summary sections. How would the user know?

5. Submit what? It is unclear that anything can be created from this main window. If the submit command is there, then information on pending, unsubmitted requests should be there as well.

6. The Identification and Tracking GUI seems to possess some of the elements that are being submitted. If this is the case, then all of the information required to make a submission should be on the same window. The label “Identification and Tracking” as it is currently used is misleading as it does not suggest to the user that it contains information or fields required for the submission of a XAR. As a matter of fact, from the current GUI design, an user would never see the ID information about the XAR before submitting it. Recommend that these two screens be redesigned such that the submission information be placed on the Primary GUI and the Summaries be placed into Dialogs or subordinate tabs.
7. On “Identification & Tracking” dialog, the label “investigator” should not be followed by one button to get the information.

   Two suggestions: either

   A) Label the button “Investigator Information”

   B) (preferred). Considering the relatively small size of these dialogs as compared to others in the application, serious consideration should be given to combining these dialogs.
8. Consistent implementation of pushbuttons on the interface.

   Example 1. “Affiliation” button seems longer than the dialog it is on.

   Example 2. “Quit” button resizes with the Dialog where as the OK & Clear buttons do not. Make sure that “Quit” is in the button box. Also, consider renaming the “Quit” button to make it consist with other ECS buttons possessing the same functionality (like “cancel”).
9. “Functions: XAR Submission - XAR Status - Timeline Schedule.” No visual cue on GUI to indicate which function has been selected.

b. Compatibility with the workflow diagram for DAR Function 1.0: Specific concerns regarding the compatibility of DAR Tool main window and the “Identification & Tracking” dialog with the applicable workflow diagrams are shown in Table 5-1. This table organizes each specific concern around the specific workflow element identification number employed in the workflow analysis.

5.1.1.2.2 Area of Interest Definition & Coverage Dialog

The ‘Area of Interest Definition & Coverage dialog’ is known as the ‘*Area of Interest and Map Display screen*’ in Appendix A.

a. Human factors engineering comments:

1. Assumption: That mapping and AOI selection occurs in the dialog that is deployed when the “Area of Interest Definition & Coverage” icon is selected.

Table 5-1. Compatibility of the DAR Tool Main Window and the "Identification & Tracking" Dialog with the Workflow Diagrams for DAR Function 1.0

Workflow Identification Number(s)	Workflow Compatibility Concern
1.1.1	Unsure where to start to create a XAR. Assumption is that user chooses "XAR Submission" from "Functions." Two problems with this: A) "DAR Submission" is one of the choices under "Function." It is unclear from title whether by choosing "DAR Submission" if they are submitting a XAR, editing a XAR, or creating a XAR. B) No environmental support from the GUI as to what to do after "DAR submission" is requested.
1.1.2 - 1.1.6	The workflow suggests that the GUI should enable the user to choose between creating a new XAR, loading a XAR that is locally stored, or querying a XAR database for a XAR. No clear mechanisms are available to users to perform any of these as the GUI is currently implemented.
1.1.7 -1.1.9, 1.1.11	All the functionality suggested by the workflow is present in the GUI in the Identification and Tracking Dialog that is invoked when the "ID" icon is depressed. However, it is not going to be apparent to a user to look under an "ID" icon to find XAR submissions. It is highly recommended that all functionality related to XAR creation, editing, and submission be placed in the same primary window, if possible.
1.1.10	No way to set "ground campaign" from GUI.
1.1.12-1.1.15	No apparent way to view, set, or modify the implementation urgency of a XAR.
2.1	It does not appear that a Query of the XAR database can be performed from this interface.
1.1.16 - 1.1.23	These functions depend upon user query of the XAR database. A user can observe XAR ID, XAR Type, XAR Title, Requester ID, date & time of original submission, date & time of latest submission, investigation classification, scientific objective, ground campaign and implementation urgency of a XAR in the database. None of these can be observed if the database cannot be queried. Also, there is no indication that user has choice about modifying XAR, no obvious way to copy the contents of an existing XAR to another file, and no obvious way to select XAR parameters from another active XAR.
1.1.25 - 1.1.29	These functions also depend upon the ability of the user to query the XAR db. There is no obvious place to observe XAR data and no control to suspend an active XAR or enable a suspended XAR.

1. Button (Options List) labeled “Map Button” probably should not be an options list. The choices are not mutually exclusive. The ability to select more than one map overlay, but not all of them should be provided for. The current implementation allows only for the selection of one overlay or all of them, but not combinations.
2. Area of Interest polygon: Currently the user can draw a polygon in “space.”
3. Area of Interest polygon does not follow the map if the map orientation is changed. This is further complicated by the fact that the coordinates list for the polygon retains the original coordinates (those prior to the reorientation of the map) and never updates them.
4. What is the difference between “Delete Selection” & “Clear Selection”
5. Do the “Delete Selection,” “Clear Selection” and “Reset Display” refer to the AOI polygons or the Zoom feature. How would the user know? If it is the AOI, can you only change the last of multiple AOIs.

b. Compatibility with the workflow diagram for DAR Function 1.0: Specific concerns regarding the compatibility of DAR Area of Interest Definition & Coverage Dialog with the applicable workflow diagrams are shown in Table 5-2. This table organizes each specific concern around the specific workflow element identification number employed in the workflow analysis.

Table 5-2. Compatibility of the DAR Area of Interest Definition & Coverage Dialog with the Workflow Diagrams for DAR Function 1.0

Workflow Identification Number(s)	Workflow Compatibility Concern
1.2.4	No apparent functionality to modify existing AOIs.
1.2.7	User is unable to set AOI by selecting coordinates.
1.2.8	Start and stop times can be set from the “Observation Timing Requirements” dialog.
1.2.12 -1.2.13	View swath information is modified in the “Observation Timing Requirements” dialog.
1.2.20	This version of the DART allows users to draw a rectangle to which the main map will zoom. However, the only control that the user has over this zoom function is how large they draw the rectangle. A mouse operation may be usefull sometimes but may lack the precision for all jobs. Recommend adding an area where the coordinates of the upper left hand corner and lower right hand corner can be entered. *The functionality is present as required by the workflow but the labels by which to select the desired functionality are not clear on the GUI.*
1.2.21 - 1.2.25	No orbit functions are available from this GUI.
1.2.26	The only coordinates for AOIs that are displayed are for a square and not for any other polygon. According to the workflow this is insufficient. User must be able to enter polygon coordinates.
1.2.35 - 1.2.38	No place to enter number of points for polygon and lat/longs for points.
1.2.39 - 1.2.52	At this time, functionality to create a polygon by entering point information or editing/ adding points to an existing user defined polygon does not seem to be supported.
1.2.58-1.2.59	No evident mechanism to set or change “coverage amounts.”

5.1.1.2.3 "Observation Temporal Requirements" GUI

The 'Observation Temporal Requirements GUI' is known as the '*Observation Timing Requirements screen*' in Appendix A.

a. Human factors engineering comments:

1. Assumption: most of the "Observation Timing" functionality is possessed under windows opened by the icon labeled "Observation Temporal Requirements."
2. The same type of widgets used to enter start and stop times should also be used to enter acquisition window times.
3. Button labeled "further swath options." It was never clear on this interface that you were manipulating swath options to start with. Additional labeling should be provided on the GUI to let user know what the available functions on the GUI or at least what functional area the interface affects.
4. Push Buttons & Options Buttons. Button labels should make correct use of letter case rules.
5. In general, the widgets on this interface should be aligned. BX possesses an alignment editor that is not difficult to use.
6. If "Further Swath Options" continues to remain as its own dialog, change the title that appears in the window header. It currently reads "darObsSwathDialogShell."
7. Do not use sliders in the "darObsSwathDialogShell." The current scale is from 0 to 99,999. Currently the smallest increment by which the scalar will move in a mouse operation is 360 and it takes minutes to move the scalar even small distances with the key board arrow keys. Within the space on a UNIX workstation display, the scale widget can not be stretched long enough so that it can be advanced one unit at a time. Use a switch box instead.
8. Investigate combining "darObsSwathDialogShell" with the "Observation Timing Requirements" dialog.

b. Compatibility with the workflow diagram for DAR Function 1.0: Specific concerns regarding the compatibility of DAR "Observation Temporal Requirements" GUI with the applicable workflow diagrams are shown in Table 5-3. This table organizes each specific concern around the specific workflow element identification number employed in the workflow analysis.

Table 5-3. Compatibility of the DAR "Observation Temporal Requirements" GUI with the Workflow Diagrams for DAR Function 1.0

Workflow Identification Number(s)	Workflow Compatibility Concern
1.3.1	Assume this means selection of "Observation Temporal Requirements." Either here or on the main GUI it should be indicated whether this is being selected for a new XAR or to modify an existing XAR. If modify an existing XAR, the interface should indicate which XAR to help minimize the likelihood of the wrong XAR being altered.
1.3.2 - 1.3.2	No apparent mechanism to set "multi-observations flag."
1.3.7	Acquisition Window cannot be set with a course scalar or fine incrementer, only by a text field entry. Using a text field here is inappropriate as multiple metaphors for the same function are present on the same screen. The "acceptable start time" and " acceptable end time" are set through use of switch boxes, but the acquisition window is not.
1.3.8-1.3.9	No apparent way to set "repeat interval" duration. Is this what is meant by label "Number of Intervals?" Not clear if the slider that is present on the interface (labeled "number of intervals) sets number of repeat intervals.
1.3.10	Present on dialog that pops up when "Further Swath Options" is selected. Unclear as to why this functionality is on a dialog. It should be included on the "Observations Timing Requirements" dialog. Serious consideration should be given here and in other areas to compress the number of dialogs used.
1.3.11-1.3.16	No mechanism present to pick a "specific observation time."

5.1.1.2.4 Telescopes and Configuration Dialog

The 'Telescope and Configuration dialog' is known as the '*Telescopes Configuration screen*' in Appendix A.

a. Human factors engineering comments:

1. Most of the functionality that can be captured under a description of "instrument configuration" is implemented in the prototype within the dialog deployed by selection of the icon labeled "Telescopes and Configuration."
2. Align widgets
3. Use correct & consistent cases on widget labels
4. Look at DPR implementation of "telescope settings" to ensure consistency across the program.
5. For the VNIR telescope options, don't use an options button. A selection of toggle buttons would be better. By placing all the possible combinations an "Options List," the user may potentially read all the choices to make selection. A group of check boxes would be a better implementation The user could then pick the settings they need rather then read a list of all the possible combinations.

6. Do the SWIR & TIR telescopes have bands that DART user should be concerned with (they must be set in the DPR)?

7. What is the difference between "low gain 1" & "low gain 2"? Can this be easily conveyed to the user?

b. Compatibility with the workflow diagram for DAR Function 1.0: Specific concerns regarding the compatibility of DAR Telescopes and Configuration Dialog with the applicable workflow diagrams are shown in Table 5-4. This table organizes each specific concern around the specific workflow element identification number employed in the workflow analysis.

Table 5-4. Compatibility of the DAR Telescopes and Configuration Dialog with the Workflow Diagrams for DAR Function 1.0

Workflow Identification Number(s)	Workflow Compatibility Concern
1.4.1	No place to make choice to accept default mode.
1.4.2	No place to "set instrument mode."
1.4.4	Gain settings can be selected but the selection mechanism is not consistent with that currently under consideration for DPR for making telescope settings.

5.1.1.2.5 Instrument/Target Geometry Dialog

The 'Instrument/Target Geometry dialog' is known as the '*Instrument/Target Viewing Geometry screen*' in Appendix A.

a. Human factors engineering comments:

1. Most of the functionality that can be captured under a description of "viewing geometry components" is implemented in the prototype within the dialog deployed by selection of the icon labeled "Instrument/Target Geometry."

2. Use correct & consistent cases on widget labels.

3. Look-angle and View-swath should probably be presented together since the choices made about these occur together. For the same reason, the setting of minimum look angle, maximum look angle, day/night setting, and sun angle settings should occur together. This GUI seems to possess the min/max look angle setting, day/night setting, and min/max sun angle settings (1.5.7 - 1.5.11) but does not possess the ability to set specific look angle or swath view (1.5.2-1.5.6).

4. Look Angle button (Options box) should not contain both telescopes and views on the same list.

b. Compatibility with the workflow diagram for DAR Function 1.0: no issues recorded.

5.1.1.2.6 Cloud Coverage Requirements Dialog

The ‘Cloud Coverage Requirements dialog’ is known as the ‘*Data Quality Requirements Screen*’ in Appendix A.

a. Human factors engineering comments:

1. Most of the functionality that can be captured under a description of “data quality components” is implemented in the prototype within the dialog deployed by selection of the icon labeled “Data Quality Requirements”
2. Use correct & consistent cases on widget labels
3. Under “Seasonal Requirements,” don’t list the choices as “season1, season2, season3, season4, & any.” These names have no meaning or referent for the user. Use the names: Summer, Autumn, Winter, & Spring. “Any” should probably be changed to “None” or “No Requirements.” Also, these choices should be implemented via checkboxes so that users can choose combinations.

b. Compatibility with the workflow diagram for DAR Function 1.0: Specific concerns regarding the compatibility of DAR Cloud Coverage Requirements Dialog with the applicable workflow diagrams are shown in Table 5-5. This table organizes each specific concern around the specific workflow element identification number employed in the workflow analysis.

Table 5-5. Compatibility of the DAR Cloud Coverage Requirements Dialog with the Workflow Diagrams for DAR Function 1.0

Workflow Identification Number(s)	Workflow Compatibility Concern
1.6.4-1.6.5	Only if user selects “no” on the radio box “Avoid Clouds” should the user be allowed to select maximum acceptable cloud cover. This choice should be grayed-out until the “No” is selected.

5.1.1.2.7 Data Transmission Requirements Dialog

a. Human factors engineering comments:

1. Most of the functionality that can be captured under a description of “miscellaneous components” is implemented in the prototype within the dialog deployed by selection of the icon labeled “Data Transmission Requirements.”
2. Consider areas with which the functionality in this dialog box can be combined.

b. Compatibility with the workflow diagram for DAR Function 1.0: no issues recorded.

5.1.1.2.8 Workflows for DAR Function 1.0 not Currently Implemented in the DAR Prototype GUIs

The following detailed functions have not been implemented in the current DAR prototype GUIs to the extent identified in Table 5-.6. It is anticipated that the functionality required to implement these functions will be added in future efforts to complete the DAR Tool.

Table 5-6. DAR Workflows not Currently Implemented in the DAR Prototype GUIs

Workflow Identification Number(s)	Workflow Compatibility Concern
1.8	Item 1.8 of the workflow diagram describes functionality that allows the user to select DPR processing options for a XAR. This includes allowing the user to launch the DPR Tool from the main DART screen and create/submit a DPR for an active, new XAR. Recommend investigating an implementation solution for this function.
1.9	Item 1.9 of the workflow diagram describes functionality that allows the user to save a copy of their XAR configuration parameters for a work in progress using local storage. Recommend investigating an implementation solution for this function.
1.10	TBD – algorithm necessary to generate resource estimate not available until 7/1/96.
1.11	The ability to revise XAR parameters may be accommodated by the prototype GUIs, but it is not apparent. Recommend investigating an implementation solution for this function.
1.12	Assumed that by pressing “Submit” on the main GUI, a new XAR is created or an existing XAR is modified by the actions of all the previously discussed dialogs.
1.12.4	Functionality is present, but it should be presented on a standard BX message dialog as available from BX’s palette.
1.12.5 - 1.12.6	Does not appear to have functionality to save local copy of XAR configuration parameters or to trouble shoot iaw written procedures.

5.1.1.3 HFE Recommendations for Enhancing the DAR Tool Prototype GUIs and Dialogs

The preceding sections listed specific comments and deficiencies of the DAR prototype based on the workflow analysis and general Human Factors considerations. The following comments are more global, intended to address conceptual implementation of the prototype. Five specific areas are addressed.

5.1.1.3.1 Navigation Model Enhancements

The most desirable navigation design for a given application through its GUIs is one in which the user can go through the minimum amount of steps necessary to perform critical functions. One of the purposes of workflow analysis is to help GUI developers identify optimal paths for critical and/or frequently used functions so that screen design can facilitate the user’s needs.

Good GUI design includes mechanisms of navigation which help maintain user awareness & focus. A GUI should lend environmental support to a user to keep that user aware of the task and when possible, prompt the user through critical paths. One of the primary deficiencies of the DAR prototype as it is currently implemented is that it does little to support user awareness and focus through its most critical functions; that is, the creation of new XARs and the modification of existing DARs.

It is very difficult to identify from the main DAR interface what steps are required to initiate the process of creating a new XAR. It was impossible to identify from the main interface what steps were necessary to modify an existing XAR. Clear identification should take place on the main interface and subsequent dialogs that identifies that a new XAR is being created or that identifies an existing XAR that is being modified.

Another overall deficiency within the prototype is that the GUI does not support user choices and "guide" the user. For example: the main screen never prompts the user to choose between creating a new XAR and modifying an old one. Obviously, there are steps for each of these choices which are not shared by the other choice. To create a new XAR, a user may start making desired settings or may wish to copy the settings from an existing XAR. To modify an old XAR, the user must load information about the existing XAR. Given these two different paths, the user may wish to put widgets and functions applicable to both on the same interface, but gray out the widgets that are not related to the users previous choices and actions.

By graying out irrelevant choices and actions, the user is "guided" through relevant ones.

There are several areas in this workflow which call for users to make choices which are not supported within the current prototype design. Consideration should be given to providing clear mechanisms by which the user can make necessary choices and to providing clear functional paths as the outcome for each user choice.

Specific suggestions to improve overall GUI navigation include:

1. Icons on the Icon bar should be placed in the same sequence as their corresponding functionality can be found in the workflow. By placing icon buttons in the most likely order of operations, the user's need to search the icon bar for necessary functionality is reduced.

2. Provide clear distinctions between various modalities and processes. For instance: at the main screen, the choices as whether to create a new XAR; edit an existing, locally-stored XAR, or to query the database for a XAR should be a clear and distinct choice, which results in an appropriate clear path of user actions.

3. With regard to the query function, after it has been made clear how to perform a query, attention should be given to clear and navigable distinctions between different submodes within a query. Whether performing a text query; spatial query; temporal query; or a hybrid of the three, the nature of the query should be easily identifiable. The interface

should have some mechanism to declare to users what action they are performing and what data the operation is being performed on.

4. Investigation should take place into adopting a “wizard” approach to creating new XARs or editing existing XARs. It may be desirable to have a dialog that appears on top of the main window prompting the user to make a choice between various options that might include: create a new XAR, edit existing XAR, query data base, copy parameters from an existing XAR. When the user makes a selection, then further prompting would result to guide the operator through each procedure unique to each choice. Eventually, after the user has used the “wizard” to start the creation of a new XAR or the editing of an old one, the user should eventually come to one common XAR screen (or set of screens). For new XARs, common GUIs should have either no information or default information in places when information is required. Upon completion the user should be prompted as to whether new entry should be saved, and if so where to be saved. For old XARs, the fields of common GUIs would be populated by information that was already on file. To this information should result in the user being prompted as to whether or not to accept changes.

5.1.1.3.2 Screen Dialog Integration

Related to GUI navigation is the number of interfaces within an application through which a user must go to perform critical functions. As previously stated, workflow analysis is intended to help developers identify optimal paths for critical functions in GUI design. By optimizing the functional path for critical operation, generally the number of screens through which the user must navigate is reduced. By clearly defining the outcome of user choices, critical functions can be consolidated into fewer screens than designing screens that capture all the potential functionality of the system for each event or action at every step.

The following is a list of potential DAR tool “candidate” dialogs that should be considered for consolidation:

1. One DAR area that could benefit from such consolidation is under the “Observational Temporal Requirements” dialog and its “Further Swath Options” subdialog. There is no reason to make the user go through the action of selecting another dialog from the icon bar to get to functions that they will need to complete this portion observation timing settings. This same example also speaks to another point: that is the grouping of screen elements (see next section). Additionally, there are a lot of improvements that could be made to both of these dialogs that would give developers room to consolidate if space is the issue. Eliminate sliders as they are currently implemented. Some of the sliders use scales that are less than 1. The scale on other sliders start at zero and go up to 99,999. The scales are too divergent to logically be displayed together. In the latter case, the smallest increment by which a user can move the slider is 360. Use a “switch box”instead; in this

case it is more precise, takes less room, and can be manipulated by mouse operation if necessary.

2. Consideration should be given to combining the Observation Temporal Requirements dialog and the Temporal Requirements dialog.

3. Consider use of the “tab stack” widget to help consolidate related dialogs. For instance, the icons and associated dialogs “Telescopes & Configuration,” “Instrument/Target Geometry,” “Cloud, Weather, & Season Requirements,” “Calibration,” & “Data Transmission” could all be consolidated onto one secondary window, within a tab-stack widget, each former icon having its own tab. Give this secondary window an icon button with a generic enough name to cover all the functions associated underneath it, like “Telescope and Data Transmission Settings”.

5.1.1.3.3 Screen Element Groupings

One way to help reduce the number of dialogs through which a user must navigate is by functional group of screen elements. By grouping task-related functions together, the developer can reduce the users need to search for performance-critical functions. This reduces system demands to support multiple screens simultaneously and facilitates user performance.

Functions that are frequently used together should not be spread out over several dialogs. Actions that occur closely together in the users “optimal path” (as identified by workflow analysis) should be presented together to reduce the users need to search for them. Also, by presenting related user functions together on the interface it becomes easier to design application-supported guidance (such as graying-out). For a given set of functions that may be related at a programming level, several may be irrelevant or of infrequent use to a user.

The following is a list a specific DAR interfaces in which the functional grouping should be reconsidered.

1. One section where screen element grouping could be improved is in the area of the main screen and “Identification & Tracking.” All the choices related to creating a new XAR or modifying an old one should be on the same interface because the user would intuitively look for these choices to be grouped together.

2. The functional relationship of Observation Temporal Requirements and “Further Swath Options” should be reconsidered. Currently, “further swath options” is a child dialog of Observation Temporal Requirements. However, it is not clear that the functionality is subordinate to that of the parent.

3. Consider use of the “tab stack” widget to help consolidate functions. The icons and associated dialogs “Telescopes & Configuration,” “Instrument/Target Geometry,” “Cloud, Weather, & Season Requirements,” “Calibration,” & “Data Transmission” could all be consolidated onto one secondary window. Consider the fact that the user may

expect to see these functions together even though they are handled separately in programming. This functional grouping also helps in the consolidation of dialogs.

5.1.1.3.4 Timeline GUI Requirements

The following section describes an alternative “Observation Timing Tool” concept to that currently implemented in the prototype. HFE believes that the approach to the observation timing tool described in the following sections will provide the functionality desired by the client.

5.1.1.3.4.1 Observation Timing Tool Concept

In this conceptual framework, a timeline is included in the “Observation Temporal Requirements” window. This timeline may be implemented with Delphi or with independent timeline widgets that have hooks into the Delphi timeline. Upon that timeline, markers like those seen in Figure 1. would be used to set and visually illustrate start and end date/time, repeat interval duration & number, and acquisition window duration.

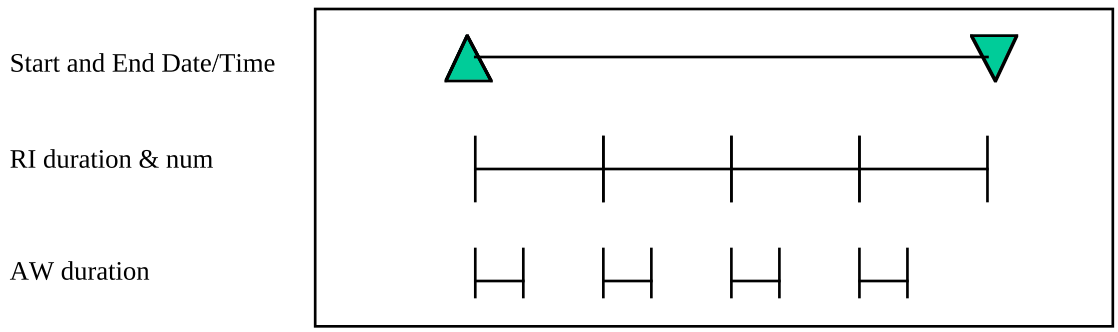


Figure 5-1. Markers to be Used on an Alternative “Observation Temporal Requirements” Interface.

NOTES:

1. The start and end date/time for acceptable data must be set before any other observation timing data can be entered.
2. Either the Repeat Interval or the Acquisition Window may be designated next, however, entering one modifies (at least initially) the other. For example, entering an AW results in an equal sized RI designation. (AWs are assigned to RIs and can be no longer than @ RI.)
3. Additional pairs of identical RI and AW combinations can be added, up to the number possible in the start/end date/time designated.
4. Number of revisits in AW should normally be designated upon completion of other observation timing parameters, since it is more meaningful at that time.

5. AW placement may be/have been deleted from the DAR API, in which case the placement of the AW always begins at with the beginning of the RI (as shown in the figure above).

Assumptions

1. The sum of all RI durations is < or = to the overall mission duration (defined by subtracting the start date/time from the end date/time.
2. RIs and AWs are of fixed durations.
3. AWs commence with the start of an RI.
4. AW durations are < or = to the duration of an RI.
5. Creating one AW prior to the creation of an RI, results in the software automatically creating one RI of equal duration (and likewise if multi-observations are toggled to Yes.
6. An implicit assumption (whose validity must be verified) is that the number of AWs monotonically increases with respect to the number of RIs.

5.1.1.3.4.2 User-Computer Interactions in Setting Observation Timings

The tool that supports this functionality could be contained in a Delphi window or as a separate (but linked) display area. One potential sequence of user-computer interactions that could be implemented is:

1. Set option to enable create/modify observation timing. (Observe observation timing display area.)
2. Decide to set start and end date/time for acceptable data.
3. Select start date/time (place start time triangle to desired date using mouse cursor and apply fine incrementer to set to exact date and time).
4. Enter specific orbit number and start date/time (using keyboard and text entry field of the format: yyyy/mm/dd - hh:mm:ss).
5. Select end date/time (place end time triangle to desired date and time using mouse cursor and apply fine incrementer).
6. Enter end date/time (using keyboard and text entry field of the format: yyyy/mm/dd - hh:mm:ss).
7. Observe highlighted start and end date/time for acceptable data.
8. Decide to accept or modify settings. (If modify do steps 3-5.)
9. Press mouse button to accept. Observe highlighting disappears.

10. Set multi-observations flag to yes/no [default: no] (Assumes yes for purpose of example.)
11. Decide to set Repeat Intervals (RIs) or Acquisition Windows (AWs). (RIs choose steps 10-11. AWs choose steps 12-13.)
12. Select RI by: (a) positioning mouse cursor to desired begin date and pressing mouse button, (b) dragging mouse cursor to desired end date/time for the RI and releasing the mouse button, (c) observing RI and deciding to accept or modify it, and (d) pressing the mouse button to accept it.
13. Enter RI duration and number (using keyboard and text entry fields for each parameter).
14. Set RI duration and number, using pick list for each parameter, where duration includes the following list of options: week, cycle, month, season, 6 months, year, other window, no requirements
15. Observe automatic placement of multiple RIs across start/end date/time and number of RIs field updated with ##s.
16. Select AW by: (a) positioning mouse cursor to desired AW begin date and pressing mouse button (within the bounds of a single RI only, (b) dragging mouse cursor to desired AW end date/time for the AW and releasing the mouse button, (c) observing AW and deciding to accept or modify it, and (d) pressing the mouse button to accept it.
17. Enter AW duration (using keyboard and text entry field).
18. Set AW duration, using pick list for each parameter, where duration includes the following list of options: week, cycle, month, season, 6 months, year, other window, no requirements
19. Observe automatic placement of multiple AWs across each RI .
20. Set number of revisits in AW (1 - n, where n is the maximum number of revisits possible.)

5.1.1.3.5 Mapping GUI Requirements

When users employ the DAR mapping tool, they are doing so to select an area from which they would like to see an image(s) of a particular location at a particular time. This location needs to be populated with a series of geographic features relevant to the selected map resolution. The user needs these features (and to toggle among those that are available) to: (1) orient themselves to a location on the map, (2) identify geographic features of relevant to the specific XAR interests, and (3) to aid in the designation of the AOI polygon itself. The main properties of this tool is not demonstrated by the prototype. For example, when the user zooms in a resolution that begins to approximate that of an “ASTER-type” view, the map turns white, containing no discernible map features whatever. To the extent that it is possible, the prototype should demonstrate the capabilities described above and operate as described in the DAR workflow.

5.1.2 Integration of the DAR Function 2.0 (XAR query) with the Release B client

The primary query tool for the Release B Client is the ESST. At present, however, the ESST is designed to support queries on archival data. Queries to the Japanese ASTER database regarding the status of active XARs is a necessary function of the Client as well. The functionality that supports XAR queries has not as of yet been defined. There are two options available to be implemented. One option is to retain the DAR query function as part of a stand alone DAR Tool (which is capable of transparently passing context to/from related client tools (e.g., context passing from the DAR Tool to the DPR Tool, as described in the next section). The other option is to allocate the DAR query function to the ESST and thus maintaining the 'one-stop shopping' paradigm that has been used in client query design to date.

From the point of view of the user, it would be more optimal to integrate the DAR query function with the ESST, so long as the added functionality does not induce unacceptable complexity into the ESST GUIs. However, this must be weighed against other factors, including cost, schedule, and technical risk, before a final decision can be made. Added considerations include the outcome of developments in the ASTER API and the degree to which the Japanese ASTER database supports the complex queries envisioned in DAR Function 2.0. Absent the necessary query capability built into the Japanese ASTER database, the Release B DAR query tool will need to be sufficiently smart to parse complex science user queries into 'chunks' that are meaningful to the Japanese ASTER database and to construct from the response, a query result that meets the needs of the science user. This degree of intelligence may result in the need to design a 'smart client', which is probably beyond that envisioned in this program to date.

Continued development is therefore required to make the final decisions required to complete the design to support the DAR query function. This will in all likelihood involve the development of prototype GUIs and assessments based on paper prototyping exercises to compare among the available design alternatives.

5.1.3 DAR/DPR Links

The ASTER science community needs the capability to transparently attach DPR(s) to a XAR. As currently designed, the DPR Tool prototype contains a series of dialogs that are intended to be launched from either the ESST or from the DAR Tool. The GUI mechanisms for accomplishing this linkage has been defined for the ESST. However, the exact GUI mechanisms for accomplishing this linkage via the DAR Tool has not been addressed at this time.

A related issue is the matter of linking DPRs to DARs via the subscriptions process. The mechanism for automatically issuing a DPR on behalf of a user has not been defined. It is expected that the ECS Design Issue Team looking into the design of the subscription server will provide the groundwork design for this mechanism.

5.1.4 Timeline

Further work on the timeline should be undertaken to extend it to provide physically meaningful information which at this point has not been determined. The examples of the information needed are how to calculate the orbit-relative times according to GMT, how to calculate the time intervals in which the instruments will cross the given areas, how to calculate the equator crossings, how to calculate the terminator crossings and how to query the databases if the related physical information is already stored in the databases. The code for the prototype either stubbed-out the above functionality, or implemented it using approximated algorithms which need to be corrected.

5.2 Design and Technical Issues

The following documents of design and technical issues associated with the ASTER DAR.

5.2.1 COTS Mapping Software

STK (Satellite Toolkit) software libraries were selected for use in the prototype for graphical display of geographical-related information, including satellite ground track, mapping of data coverage, and user selection of areas of interest for DARs. After extensive prototyping work with the STK libraries, it was decided to issue an Request for Proposal (RFP) for a package which would more fully meet the functionality it was felt was necessary in the DAR tool. It is expected that the RFP will result in the selection of a replacement for the STK mapping libraries. Further work on the graphical functionality should be done with the package selected from the RFP to avoid developing with code which will be discarded and replaced.

Since the mapping requirements for the DAR were more demanding than those of the general ESST, the COTS libraries selected for use with the DAR tool will be able to fully meet ESST mapping requirements. This would allow one software library to support all Client mapping functionality.

5.2.2 Reuse of ESST

The current DAR Tool prototype attempted to preserve the look and feel of the Release B client, in particular the ESST. The main window of the DAR Tool prototype employs the GUI paradigm used in the ESST main window. That is, the DAR main window consists of a dynamically expandable iconic toolbar and a summary display area containing a high-level depiction of textual summary results, graphical map, and timeline. This effort at GUI commonality is to be applauded because it represents an attempt to reduce GUI complexity through the reuse of the same GUI paradigm. However, the DAR Tool GUIs must integrate two disparate functions—XAR creation/submission and XAR query—in order to be effective. At present, this has resulted in the addition of a 'toggle-bar' across the top of the DAR Tool main window that preempts the operation of the toolbar that is based on the ESST. In effect, this particular solution to the requirement to support multiple DAR modalities or states (that of a XAR create state and a XAR query state) has yielded a moded GUI, for which inadequate mode information is presented by the GUI to the end-user. End-user need to be enabled by the GUI to readily identify their current mode or state and thus avoid the confusion and potential for catastrophic loss of information that might otherwise occur. A further problem of this GUI design is the added complexity associated with end-users knowing or figuring out ‘what to do next’.

5.2.3 DAR Valids Exchange with GDS

Details for formats and method of delivery from the ASTER GDS need to be addressed. The GDS controls the baseline set of DAR parameters and valids. The current baseline is for the GDS to deliver the parameters and valids to ECS. The information provided by the GDS must include parameters, valids information including valids ranges and default values, parameter dependencies, and documentation sufficient to understand delivered formats. It will be the responsibility of ECS to convert the information to an ECS format service signature, and submit it to the advertising database.

This issue was noted at the ECS GDS Interface Coordination Meeting of July 22-26, 1996. The issue is noted here to alert developers to the dependency, since changes to the parameters and valids could impact development.

5.2.4 DAR Service Signature

The DAR will store its service signature, which will be derived primarily from the GDS delivered valids and parameters, as a service signature in the advertising database ECS, through the Release B Chief Architects Office, is in the process of specifying a format for storing these service signatures. The DAR design calls for the DAR tool, upon startup, to access the advertising data base for its service signature, and use the signature to build its GUI screens. The specification for service signatures should precede significant further work on an operational DAR tool, since service signature information in the prototype has been hardcoded in the tool itself.

5.2.5 Redundancy of Schedule Information

The DAR API set provides an interface for obtaining ASTER schedules from the GDS. This API appears to be redundant with data flows which provide the same schedules to ECS. The availability of ASTER instrument schedules (Long Term Schedule, Short Term Schedule, Daily Schedule) within ECS, either with the Data Server or in the FOS Data Management database. Because of longer response times associated with the API which is serviced in Japan , as opposed to obtaining the same information from ECS sources, it is desirable to obtain the schedules from an ECS source. The GDS developers have expressed a desire to delete the API for obtaining the schedules from the specification when ECS verifies that the information can be obtained locally.

5.2.6 AOI Resource Estimation Algorithm

The ASTER science community needs to have the ability to estimate the demand for instrument resources associated with the submission of a XAR of a specified area of interest (AOI). This estimate would enable the science user to prioritize their requests based on the scientific merit of a particular XAR weighted against the cost (in AOI resources) required to obtain the data. ASTER instrumentation are fixed, finite resources which have been allocated between the U.S. and the Japanese ASTER science communities. The tentative designation of an AOI resource estimation algorithm has been received from JPL scientists and engineers representing the U.S. ASTER science community for incorporation into the DAR Tool. This algorithm could be applied by a science user prior to submitting a XAR to estimate the resources required to collect the data. The algorithm calculates the 'total Area of an AOI', which is used by the science user to assess the resource intensiveness of a given XAR.

The AOI resource estimation algorithm has two variants, one derived from the other. Equation 1 is:

$$Total\ AOI\ Coverage = (Area) * (Total\ number\ of\ repeats) \qquad \text{Eq. (1).}$$

Equation 2 simply adds a standard divisor, as shown below:

$$Total\ AOI\ Coverage = \frac{(Area) * (Total\ number\ of\ repeats)}{[Scene\ area\ (3{,}600\ sq.km.)]} \qquad \text{Eq. (2).}$$

The selection of which variant to apply in the DAR Tool is anticipated to be made by the ASTER science community in the near future.

This page intentionally left blank.

Appendix A. Data Acquisition Request Tool GUI Screens

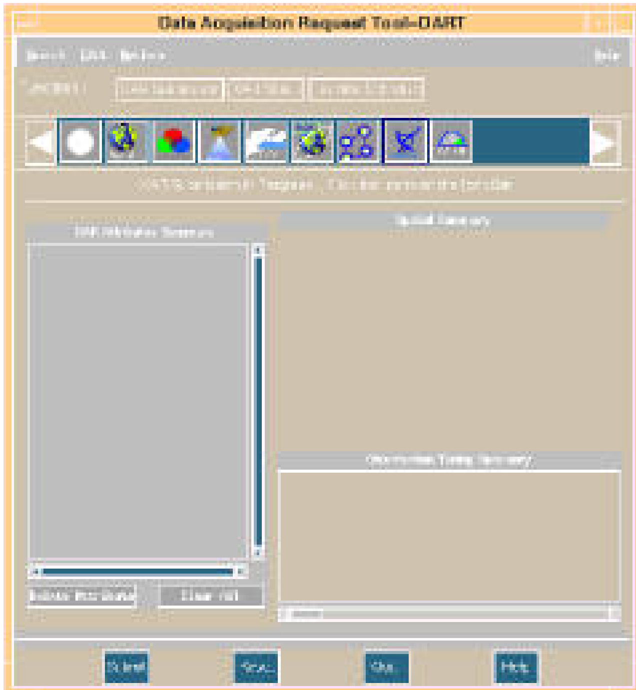
Content

Introduction

The Data Acquisition Request (DAR) prototype provides motif-style graphical user interfaces (GUIs) by which science users submit DARs from the ECS client. DARs are requests by users for scheduling data acquisitions by the Advanced Spaceborne Thermal Emission and Reflection Radiometer (ASTER) instrument. These requests are submitted by the ECS client to the ASTER Ground Data System (GDS), located in Japan. The ASTER GDS controls scheduling of the ASTER instrument, and provides the collected data as level 1A and level 1B data to the EROS Data Center.

The screens have limited functionality and are intended primarily to provide coded examples of screen layouts for the ASTER DARs. There is no "back-end" to the prototypes (i.e., the GUIs do not provide connections to any external servers for processing user-entered information). "Valids checking" of user inputs is not systematically provided for all user-input fields. Note: The screen captures are not image mapped and are meant only as visual representations of actual DAR screens; they are not to scale.

Initial Screen



When the initial DART (Data Acquisition Request Tool) screen comes up, a line of white text appears that says "Please select one function on the functions tool bar." Three options are available at this point: create a XAR, query status of existing XARs, and query timelines of existing XARs.

To create a XAR, click the XAR submission button in the functions tool bar. This brings up a warning message box that can be closed by clicking OK after reading it. Now the screen is ready to create a XAR. This is done by selecting attributes from the attribute tool bar to customize the request. When the Submit button is clicked, the DAR ID from the ICC Screen appears.

To query status of existing XARs, click the XAR Status button in the functions tool bar. This brings up a warning message box that can be closed by clicking OK after reading it. Now the screen is ready to query a XAR. Define the XAR to be queried by selecting attributes from the attribute tool bar to customize the request for status. When the Submit button is clicked, the XAR Status Screen appears.

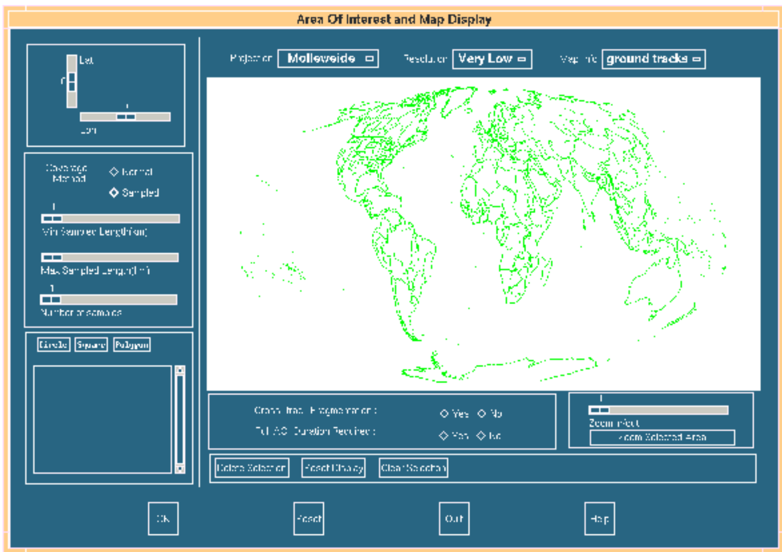
To query timelines of existing XARs, click the Timeline Schedule button in the functions tool bar. This brings up a warning message box that can be closed by clicking OK after reading it. Now the screen is ready to query a timeline. This is done by selecting attributes from the attribute tool bar to customize the request. When the submit button is clicked, the DAR Prototype Timeline Screen appears.

Attribute Screens

Identification and Classification Screen

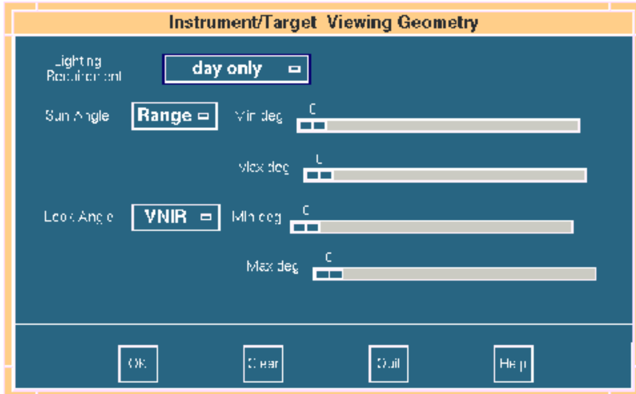
The Identification and Classification Screen allows the user to specify the type of query to be created. For XAR type, depress the desired button in front of the type of request to be made (e.g., DAR, STAR, or ETR). Instrument, Implementation Urgency, and Investigation Class are all pop-up menus with the valid values listed. Click on the desired value. To specify investigator, click on the Further Information bar and a text field window will appear that needs to be completed. When all of the information is correct, click the OK button at the bottom of the screen.

Area of Interest Definition and Coverage Screen



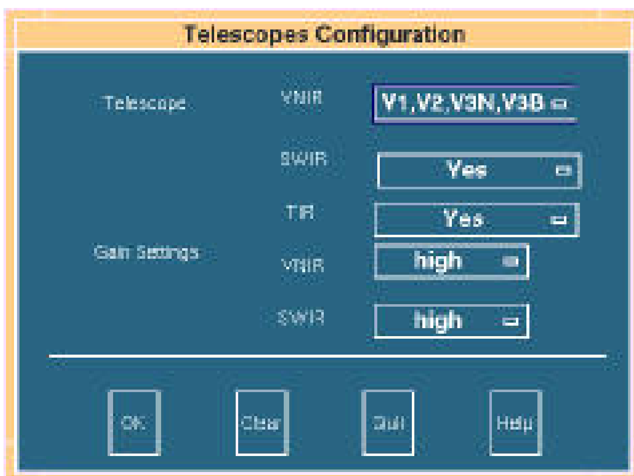
The Area of Interest Definition and Coverage Screen allows the user to define area of interest and specify coverage criteria such as sampling, cross track fragmentation and area of interest duration for the query. Map navigation is supported by setting the desired values for map projection, latitude/longitude of the center of the map, zoom factor and map resolution. This is done by making selections through pop-up menus for map projection and resolution or by clicking on the sliding scale for latitude/longitude and zoom values. When all of the information is entered, click the OK button at the bottom of the screen.

Instrument/Target Viewing Geometry Screen



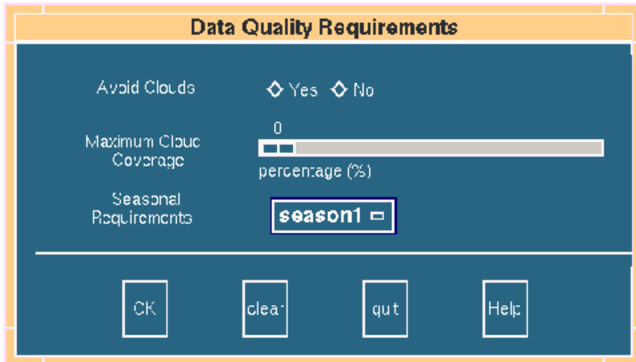
The Instrument/Target Viewing Geometry Screen allows the user to specify the Lighting Requirement, Sun Angle, and Look Angle for the query. All three choices have pop-up menus with the valid choices listed. For the Sun Angle and Look Angle, minimum and maximum degree angles can be specified. This is done by clicking on the pointer box and sliding it back or forth to the desired range. When all of the information is correct, click the OK button at the bottom of the screen.

Telescopes Configuration Screen



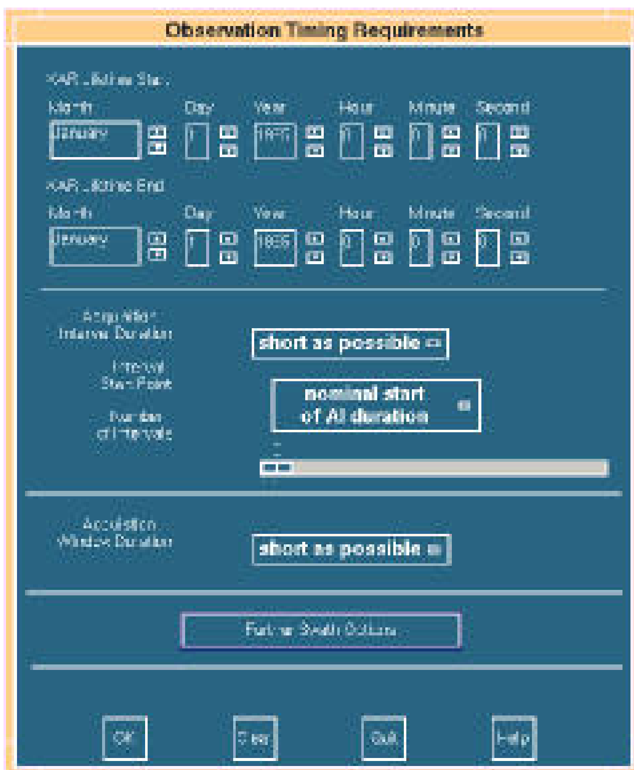
The Telescopes Configuration Screen consists of pop-up menus for Telescope and Gain Settings. Click on the desired values for each menu. When all of the information is correct, click the OK button at the bottom of the screen.

Data Quality Requirements Screen



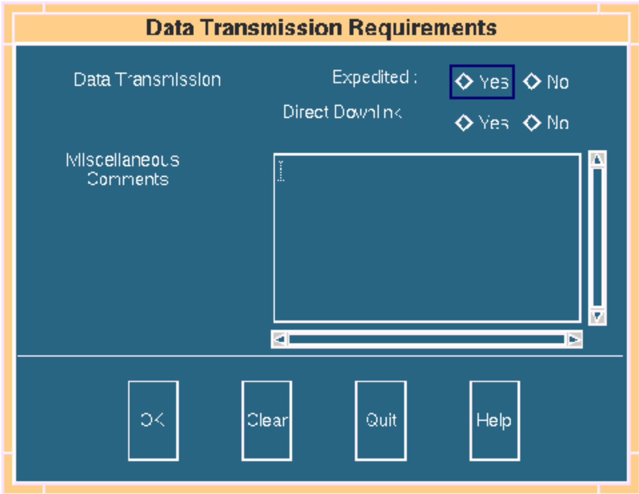
The Data Quality Requirements Screen consists of a push-button choice for avoiding clouds or not, a sliding bar for Maximum Cloud Coverage, and a pop-up menu for Seasonal Requirements. When all of the information is correct, click the OK button at the bottom of the screen.

Observation Timing Requirements Screen



The Observation on Timing requirements screen allows the user to specify time considerations for the query. The screen supports incremental date setting and time fields to set XAR lifetime values. Pop-up selection boxes are supported for values involving "durations". The query can be further fine-tuned by clicking on "Further Swath Options" and setting values for the given options using the slide bar provided. When all of the information is entered, click the OK button at the bottom of the screen.

Data Transmission Requirements Screen

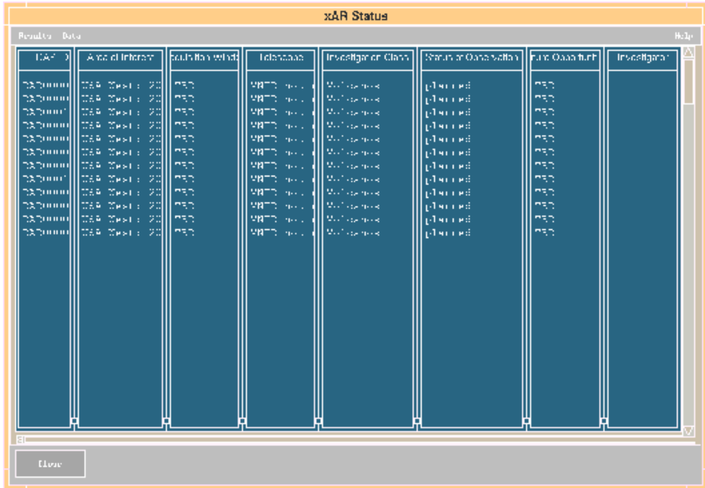


The Data Transmission Requirements Screen consists of Yes/No choices for Data Transmission Expedited and Data Transmission Direct Downlink. There is also a text field for any miscellaneous comments or to specify other requests. When all of the information is correct, click the OK button at the bottom of the screen.

DAR ID from ICC Screen

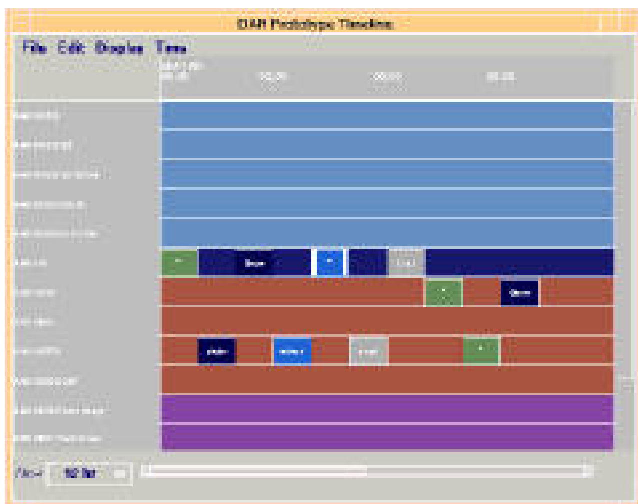
This is mainly a confirmation screen. It tells you the ICC Assigned DAR ID number and the DAR Version number. Click OK when the information has been read.

XAR Status Screen



This screen is used to view the current status of the existing XARs and contains information such as DAR ID, area of interest, investigation class and status of observation. When all of the information is viewed, click the Close button at the bottom of the screen.

DAR Prototype Timeline Screen



This screen is used to view the current timelines of the existing XARs. Clicking on the show button lets the user change the time frame of the timelines on the order of minutes to weeks.

This page intentionally left blank.

Appendix B. Development Notes for Timeline Software

Acknowledgment

The help from the FOS group is greatly appreciated. They provided the executable standalone preliminary timeline and the related source codes.

Overview

Hughes class libraries in Delphi™ provide the timeline functions both in MOTIF and Xview. We only address MOTIF applications here. If a class name begins at Cs, this class is a Client subSystem class which is derived from the classes in Delphi™.

The primary timeline class is CsTlXtl. The management class between your application and the timeline is CsTlAppl. If your purpose is to apply the timeline in your application, you needn’t know any classes except the class CsTlAppl. When it is used as part of your application, you can simply aggregate the application-specific class CsTlAppl. The way to aggregate depends on your requirements. How to create a timeline application is described in Chapter 4.

The timeline layout created by CsTlXtl is illustrated below:

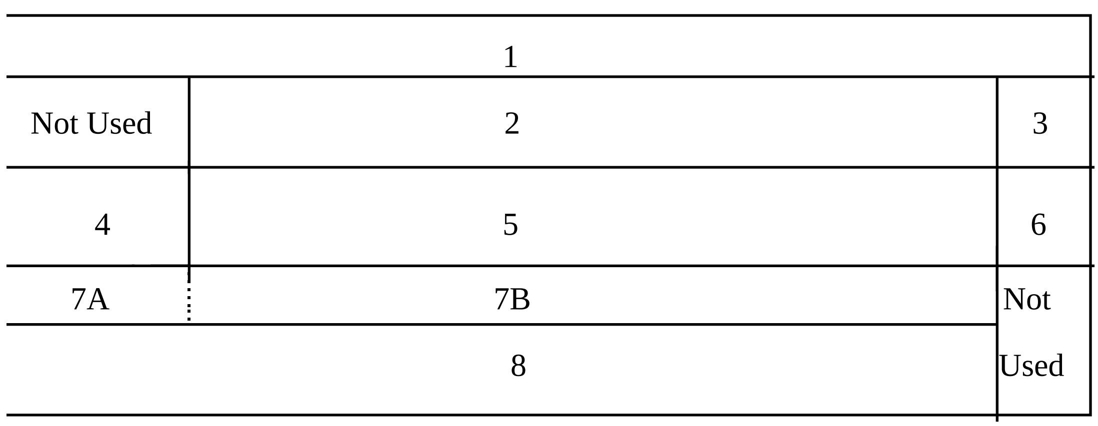


Figure B-1 The timeline layout

The timeline frame is divided into 10 areas. Among them, 8 areas are used. CsTlXtl creates the above areas and places them on a specific frame in a specified area. CsTlXtl also maintains coordination between the areas when operations, such as scrolling, require actions from more than one area.

Area 1 is the menu bar. The class is named as CsTlPanel.

Area 2 is the time scale. The class is named as CsTlTimeScale.

Area 3 is the time scale unit. The class is named as CsTsUnitRegion.

Area 4 is the label region. The class is named as TLLabelRegion.

Area 5 is the main region. The class is named as CsTlMainRegion.

Area 6 is the vertical zoom bar. The class is named as TLRsScroll.

Area 7 is the horizontal zoom bar. The class is named as CsTlZmBar.

Area 8 is the status bar. The class is named as CsTlStatusBar.

An executable timeline and its environments have been created and put in the directory /home/ywang/dar/tl/bin. The functionality of each class will be perceived by running st_tl in this directory.

Running the Demo

Enter the directory /home/ywang/dar/tl/bin. **Note: Access to the timeline code and standalone demo is restricted. For details on accessing the timeline, contact the author of this paper**.

Run st_tl

This demo has the capability

- to display the instrument and AM-1 spacecraft activities and related information chronologically on a schedule timeline;
- to display any user-defined combination of the activities and useful messages such as equator crossings, terminator crossings and mode changes etc.;
- to display textual information on any activities;
- to display time in GMT, local solar time or orbit-relative times;
- to display the time interval of an event and the time interval between the neighboring events;
- to filter observation data displayed on a schedule timeline;
- to allow the user to control the scale of the timeline;
- to allow the user to continuously scroll through the timeline in two directions;
- to allow the user to know which events take place at a given time instant.

The functionality of each area shown in Figure 1-1 is addressed below.

Area 1 CsTlPanel

File->Load Configuration

Load a configuration file.

File->Save Configuration

Save the current configuration to a file.

File->Exit

Exit from the timeline.

Display->Filter events

Make the events visible or invisible. The events which falls in the selected types or in the selected XAR IDs will be invisible. Multiple selections are acceptable.

Display->Change interval probes

This menu button is used to edit the interval probes or implement the functionality in classes CsHIntervalProbe and CsTlIntervalProbeView. Since we haven't determined whether to allow users to edit their XAR requests in the timeline, this menu button performs nothing at this stage.

Display->Change colors

Reconfigure colors of sub-regions and events. It takes a much longer time to initialize the color buttons when this menu button is first selected. Multiple selections are not acceptable. The color bar for the selected type is displayed between the color buttons and the type list. The color of the color bar will be changed by pressing a color button. The color of sub-regions and events in the selected type will be changed by pressing the button Apply or Ok. A type is selected by shifting the cursor to the type and then pressing left button in the type list.

Display->Change resources

The viewed resources can be any user-defined combination of all of the available resources.

Time->Change plan window

The visible time interval is a part of the plan window. The start time and the end time of the plan window can be changed.

Time->Change probes

Delete the existing time probes and create new time probes. The default time for the new time probe is the center of the visible time interval. A time probe can be slaved to timer.

Multiple selections are not acceptable.

Time->Time Scale: Orbit

Display time in orbit-relative times.

Time->Time Scale: GMT

Display time in GMT.

Time->Time Scale : Local Solar

Display time in local solar time.

Edit->Edit Selected

Not implemented yet. This menu button may be removed if the user is not allowed to edit events.

Edit->Unallocate Selected

Not implemented yet. This menu button may be removed if the user is not allowed to edit events.

Edit->Lock Selected

Not implemented yet. This menu button may be removed if the user is not allowed to edit events.

Edit->Unlock Selected

Not implemented yet. This menu button may be removed if the user is not allowed to edit events.

Edit->modeMenu

Not implemented yet. This menu button may be removed if the user is not allowed to edit events.

Area 2 CsTlTimeScale

The major and minor ticks, the dates and the times in GMT or local solar time(Cycles/Numbers and fractional part in orbit-relative times) are displayed. When the time probes exist, the time instants to these probes will be displayed with boxes.

The X events implemented can fulfill

drag a time probe

If a time probe exists, the time probe will keep shifting by moving the mouse cursor to the middle of the time box, holding the left mouse button and continuing moving the mouse cursor. The time probe will drop to the new position when the left mouse button is released. The state or activities crossing this time instant can be viewed by scrolling vertically.

Area 3 CsTsUnitRegion

Display the text string GMT, LST or ORT. From this region, we'll know whether the time is displayed in GMT, local solar time or orbit-relative times. It is supplementary to other regions.

No X events are implemented.

Area 4 TLLabelRegion

Display the labels of the sub-regions.

No X events are implemented.

Area 5   CsTlMainRegion

This is the main region. The timeline data is actually displayed in this region. Box views are currently provided for the resource state or activities versus time. The time probes are displayed as vertical lines in this region.

The X events implemented can fulfill

Display textual information

Double click left mouse button on an event, a window pops up to display the textual information related to this event.

Display the time interval of an event

Move the mouse into an event, the time interval of this event will be displayed in Area 8.

Display the time interval between the two neighboring events

Press the left mouse button in an open area(no events) of this region, the start time of this area which is the end of the event on the left and the end time of this area which is the start of the event on the right will be displayed in Area 8. If there is no event on the left within the plan window, the text string "indefinit" is displayed as the start time, and the same rule applies to the end time.

Select an event

Press the left mouse button on an event. The left handle and the right handle of the selected event are displayed and the event is selected.

Drag an event

An event can be switched to different resources or shifted to another location. Move the mouse cursor on an event(not inside handles), hold the left mouse button to drag. When the left button is released, the event will be put at the new location.

Resize an event

Select an event first, move the cursor to the middle of either the left handle or the right handle and then hold left mouse button to drag the mouse.

Area 6 TLRsScroll

Manages which subsets of the total vertical(resource) information is currently being displayed in Area 5. The number of visible resources(sub-regions) is determined by the scroll bar size.

X events implemented can fulfill

Resize the scroll bar

Hold the left mouse button and shift the top edge or the bottom edge of the scroll bar. The size of the scroll bar will be changed after the left mouse button is released.

Scroll vertically

Click the left mouse button when the cursor is outside the scrollbar.

Area 7 CsTlZmBar

The duration and the start time of the visible time interval and be changed.

X events implemented can fulfill

Change the visible time

Press the button in which the current time interval is shown and then select a new time interval from the option menu.

Scroll horizontally

Hold the left mouse button and drag the scroll bar. The start time and the end time of the new time interval will be displayed in a slider.

Area 8 CsTlStatusBar

Display a line of textual information. It is supplementary to other regions.

No X events are implemented.

Directory Structure and Configuration of Prototype Software

Using an HTML browser, open the file /home/pscms/delphi/index.html. The instructions and class hierarchies of Delphi will be seen.

Often used Delphi directories

/home/pscms/delphi/hcl/midsp ... classes wrapping X/Motif

/home/pscms/delphi/hcl/misc utility classes

/home/pscms/delphi/tcl/xtl ... classes for the timeline

/home/pscms/delphi/tcl/xtlExt .... the timeline with configuration

/home/pscms/delphi/rcl/rcl ... resource class library

The source code provided by FOS is in

/home/ywang/fos/tl

Timeline software is in my directory

/home/ywang/dar/tl/src

/home/ywang/dar/tl/include

/home/ywang/dar/tl/bin

/home/ywang/dar/tl/db

st_tl is the latest timeline.

st_2 is the old one.

In the following directory is a timeline application built with Builder Accessory:

/home/ywang/ep7/dar/tl

Form.cxx is the example to call the timeline in the application built by BX. This test of BX used an old version of the software.. Since then, the software was modified extensively. To run this test, would require the addition of many new classes must be added to the makefile in this located in the directory home/ywang/dar/tl.

Creating a Timeline Application

The timeline frame can be created by the member function create() or createAppl(…) in the class CsTlAppl depending on whether the timeline is standalone.

When the timeline is standalone, the member function create() will create a TopLevelShell widget and a XmForm widget as the timeline frame. There are no arguments in this function.

The example is shown in /home/ywang/dar/tl/src/etlMain.cc.

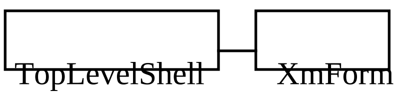


Figure 4-1 The frame for the standalone timeline

When the timeline is non-standalone, the member function createAppl(…) will create a PopupShell widget and a XmForm widget as the timeline frame. The arguments are as the same as in the X/Motif function XtCreatePopupShell(…) . createAppl() simply calls this X/Motif function and wraps the created shell widget into the class DWindow. If no resources are provided, the timeline frame will be put in a default position.

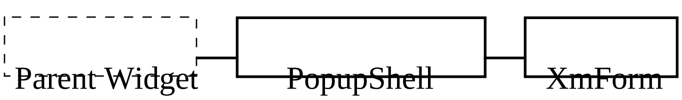


Figure 4-2 The frame for the non-standalone timeline

When the timeline has its own event loop, the example is shown in /home/ywang/ep7/dar/tl/Form.cxx. In this case, the timeline application can be initialized just before it is invoked. See the member function Form::timeLineCBCallBack(). When it doesn't have its own event loop, the example is also in home/ywang/ep7/dar/tl/Form.cxx. In this case, the timeline application must be initialized before your event loop starts. See the member function Form::create() where the timeline is initialized before the callback is invoked. In these two examples, an application shell is created. The upper left coordinates of the timeline can be set to any place. Or, you can simply use the default.

Besides the timeline frame, the color map is also needed. Currently, the default color map is used.

Configuration

The resources, events and parameters used to initialize the timeline are stored in two data files. One is /home/ywang/dar/tl/db/resources.db and the other is /home/ywang/dar/tl/db/etlConfig.db. Suppose that the first data file comes from the Control Center and the second file is defined by the user. The file names should be given in the command line. See /home/ywang/dar/tl/bin/st_tl for detail.

The formal data structures have not been defined yet. Here, we simply defined some in order to run the timeline. Any data row begins at a keyword.

The functionality of the related classes are described here.

CsTlConfig:	The base class for the rest configuration classes.
CsRcRsConfig:	Load data from the file resources.db. CsTlAppl contains a pointer to this class.
CsTlApDefsCfg:	Load and save the initial parameters for CsTlAppl objects. CsTlAppl contains a pointer to this class.
CsTlRsConfig:	Load and save the viewed resources requested by the user. CsTlAppl contains a pointer to this class.
CsTlDefsConfig:	Load and save the initial parameters for CsTlXtl objects. CsTlXtl contains a pointer to this class.
CsTlFilterConfig :	Load and save the filtered types. CsTlXtl contains a pointer to this class.

CsTlProbeDefautls: Load and save time probes. This class should be aggregated in CsTlXtl(a pointer to). So far, it has not been put in.

CsTlPtrnConfig: Load and save stipple patterns. CsTlXtl contains a pointer to this class.

CsTlTypesConfig: Load and save the colors related to resource types and event types. CsTlXtl contains a pointer to this class.

Appendix C. Guidelines for Graphical User Interface (GUI) Development Using the Human-Machine Interface (HMI) Methodology

This guideline prescribes a general approach to the preparation of the developmental Graphical User Interfaces (GUIs) for the ECS program.

Step 1: Data Input

The purpose of this step is to define the information requirements of operators/users in performing human-computer transactions using a specified ECS application to complete specific functions. The source of information requirements for this analysis varies depending upon the availability and suitability of technical documentation related to each ECS subsystem. In general, this means that if suitable (accurate and current) technical data are available in sufficient quantities, then this should be an easy step to accomplish. To the extent that there are numerous unknowns with respect to technical considerations, then this step will be progressively more difficult to accomplish and will highlight such deficiencies. Potential sources of technical documentation include:

a. Operations Scenarios for Release A and the Release B counterparts, including:

i. ECS Operations Concept, Part 2A - ECS Release (e.g., 604-CD-003-001)

ii. Operations Scenario Workshop Presentations

iii. ECS Release A Operations Scenarios (e.g., 605-CD-001-001)

b. All Release A and Release B SDPS and CSMS subsystem design specifications in which operator/user graphical user interfaces are required (one for each subsystem workflow)

(generic document numbering sequence 305-CD-0xx-00x)

c. SDPS and CSMS Requirements Specifications for Release A and B (e.g., 304-CD-002-002 & 304-CD-003-002)

d. Release A and B CSMS/SDPS Internal Interface Control Document for the ECS Project (e.g., 313-CD-004-001).

Within items b. and d., above, our interest encompasses the subsystem descriptions, object models, event traces, and use-case scenarios to the extent they are available and document information requirements for operator-computer transactions.

The approach to the identification of operator/user information requirements is to analyze the available technical documentation and, when the available documentation is not adequate, to consult any subject matter experts from whom supplementary information can be extracted. The analysis is reported in a Data Input Matrix that defines the operator/user information requirements for each ECS subsystem. Table C-1 presents a sample Data Input Matrix that was prepared for the Ingest Subsystem (INS) by the Ingest Development Team. The following paragraphs define each of the columns in the matrix and the procedures for entering data.

a. Operator/user functions

Operators/users use ECS applications to perform a given set of operations that, as a group, define a set of human-computer transactions. Human-computer transactions that yield a specified outcome are called operator/user functions. Most applications provide the software functionality that support a limited number of operator/user functions. The GUI developer needs to identify all of the operator/user functions that a given ECS subsystem application is designed to support. Each operator/user function yields an outcome (produces an observable product). Column 1 of Table C-1 identifies each of the functions for the Ingest Subsystem (INS). The numbering of each operator/user function is sequential (and may include subfunction numberings that are keyed to the operator/user function to which it belongs (e.g., 4.1 and 4.2 are both subfunctions that refer to different methods of INS control operations, as defined as Table C-1). All operator/user functions are listed in the table in this manner.

b. Objects

Since the operator/user information requirements are extracted from the object model representation of the system, the object(s) that are manipulated by or for the operator/user in performing a specified function need to be identified. This ensures the comprehensiveness of coverage of all information requirements for an operator/user to accomplish the specified function. Affected objects are listed in column 2, as illustrated by the INS objects shown in Table C-1.

c. Data Elements (Object attributes)

Each of the operator/user information requirements are identified for the object(s) that the operator/user requires to accomplish a specified function. The data elements that define the operator/user information requirements are derived from the attributes assigned to the objects listed in column 2 of Table C-1. The data elements (object attributes) are listed in column 3 of Table C-1 for each associated object and operator/user function. Only those data elements (object attributes) that form the basis of a human-computer transaction are listed in the table. This means that only those data elements that operators/users input, edit, or observe to accomplish a specified function are listed in the table.

Table C-1. Data Input Matrix of Ingest Subsystem (INS) Information Requirements (1 of 2)

Operator Function	Object	Data Element (Object Attribute)	Operator Interactions (Edit, Input, Display)
1. Media Ingest	InRequest	Media Type Total Media Count Media Volume ID Media Data Provider Delivery Record File Name Delivery Record File Directory Request ID Request State	Input Input Input Input Input Input Display Display
2. Status Monitoring	InRequest	External Data Provider/User Name Request ID Total Ingest Data Volume Request State	Input/Display Input/Display Display Display
3. Request Control	InRequest	Request ID External Data Provider/User Name Control Type list: Cancel, Hold, Resume, Change Priority New Request Priority Completion Status	Input Input Input Edit Display
4.1 Threshold Control	InSystemThreshold	Max Ingest Request Max Total Data Volume Completion Status	Edit Edit Display
4.2 Threshold Control	InExternalData ProviderThreshold	Max Number of Requests Max Total Data Volume Max Transfer Retries Polling Timer Ingest Priority Completion Status	Edit Edit Edit Edit Edit Display
5. History Log Viewing	InHistoryLog	Start Date/Time Stop Date/Time External Data Provider Data Type Final Request Status Request Id Number of Data Volume ingested Number of Granules/Datasets ingested Number of Files ingested	Input/Display Input/Display Input/Display Input Input Display Display Display Display

Table C-1. Data Input Matrix of Ingest Subsystem (INS) Information Requirements (2 of 2)

Operator Function	Object	Data Element (Object Attribute)	Operator Interactions (Edit, Input, Display)
6.1 Template Editor	InDataTypeTemplate	Data Type File Type List Completion Status	Edit Edit Display
6.2 Template Editor	InFileTypeTemplate	Data Type File Type Metadata Object Class Name Science Object Class Name Minimum Required Files Maximum Required Files ArchiveFlag RequiredFlag Completion Status	Edit Edit Edit Edit Edit Edit Edit Edit Display
6.3 Template Editor	InSourceMCF	Data Type Parameter List: Parameter Name Parameter Alias Computer Science Data Type (CSDT) CSDT Offset Completion Status	Edit Edit Edit Edit Edit Edit Display
7. User Network Ingest*	InRequest	Data Type List -File List: --File Name --File Directory --File Type Data Expiration Date/Time Desired Priority External Data Provider/User Name Delivery Record File Name Request ID Request State	Input Input Input Input Input Input Input Input Input Display Display

d. Operator Interactions

For the sake of simplicity, the types of interaction required for operators/users to interact with specific data elements are categorized into one of three classes: Edit, Input, and Display. The required data on interactions include identification of the functions and associated data elements, and specific identification of the operator/user interaction with each data element, viz., whether it is 'display only,' (i.e., data elements which are displayed but will not be changed), 'edit,' (i.e., data elements whose current value will be displayed, but may be changed by the operator/user), or 'input only,' (i.e., data elements whose current value is blank or null, and which will be entered on that occasion). These interaction types are listed in column 4 of Table C-1. Multiple entries into the table are required when operators/users interact with data elements in more than one manner.

The product of this step is the creation of a data input matrix for each ECS subsystem that is similar to the sample matrix shown in Table C-1. Each matrix should be reviewed by members of the development team and the human factors team and the final matrix prepared based on comments received.

Step 2: Workflow Analysis

A workflow analysis should be performed for all ECS applications, whether X/Motif developmental, COTS/OTS, or HTML-based application. The purpose of this step is to analyze the operator/user workflows that are required to execute each operator/user function identified during step 1. The following statement, which applies to textual material, applies equally to workflow analysis:

'Good writing is clear thinking made legible.'

The same is true for effective GUI design. If the workflow (i.e., how the operator/user conducts transactions with the system) can't be adequately defined, then it may not be feasible to develop a usable GUI.

Diagrammatic workflow analysis is a two-level graphical technique for defining and structuring the ECS subsystem functions identified in step 1 and the associated human-computer transactions that occur as a consequence of manipulating the information requirements of each subsystem. The procedures for performing a workflow analysis are described below.

a. Prepare Top-Level Function Analysis.

Review the Data Input Matrix, prepared as a part of performing step 1. Extract the operator functions listed in column one of the matrix and place them a top-level function flow diagram, as illustrated in Figure C-1. Since each ECS subsystem application will be initiated through activating an icon on the ECS desktop, which leads to the display of the application main window on the desktop, identify the top-level of the function flow, as shown in the function numbered 1.0 in Figure C-1. All operator functions that should be accessible directly by means of the main window should be displayed on the functions row, as illustrated in the figure (i.e., functions numbered 2.0 through 8.0). Each function should be connected to the top-level

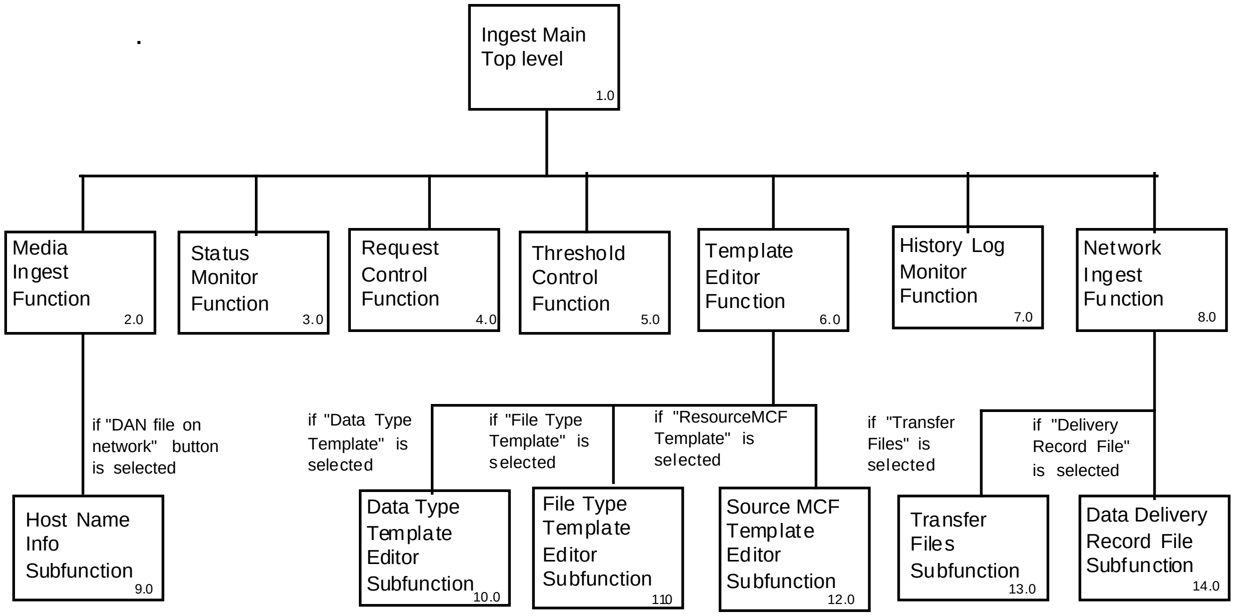


Figure C-1. Top-level Function Flow Diagram of the Ingest GUI Function

Function by means of a series of connected vectors (lines), as illustrated in the figure. All operator actions which should be accessed only through the activation of a given operator function, are called operator subfunctions. Subfunctions should be listed on the subfunction line directly beneath the operator function to which they are subtended, as illustrated in the figure. The conditions for accessing each subfunction (e.g., if "DAN file on network" button is selected) should be attached as a label to the vector that connects each subfunction to its specified operator function. See the figure for sample conditions. Once the top-level function flow diagram is complete, the developer should review the diagram for completeness and accuracy. It should be understood that this is a highly iterative process that may require the developer to revise this diagram--as well as previous and subsequent diagrams based on changes in perception regarding the optimum ordering of functions, subfunctions, and actions of operators/users.

b. Prepare Workflows for Individual Operator Functions.

Prepare a graphical workflow diagram for each operator/user function and subfunction depicted in the top-level function analysis, except for the top-level function (numbered 1.0 in Figure C-1). This graphical workflow diagram will identify a sequence of operator/user interactions with the computer required to accomplish the function. Therefore, the diagram should provide a logical ordering of the human-computer transactions that occur, as illustrated in Figure C-2. The procedures for preparing a graphical workflow diagram of operator/user functions are described below.

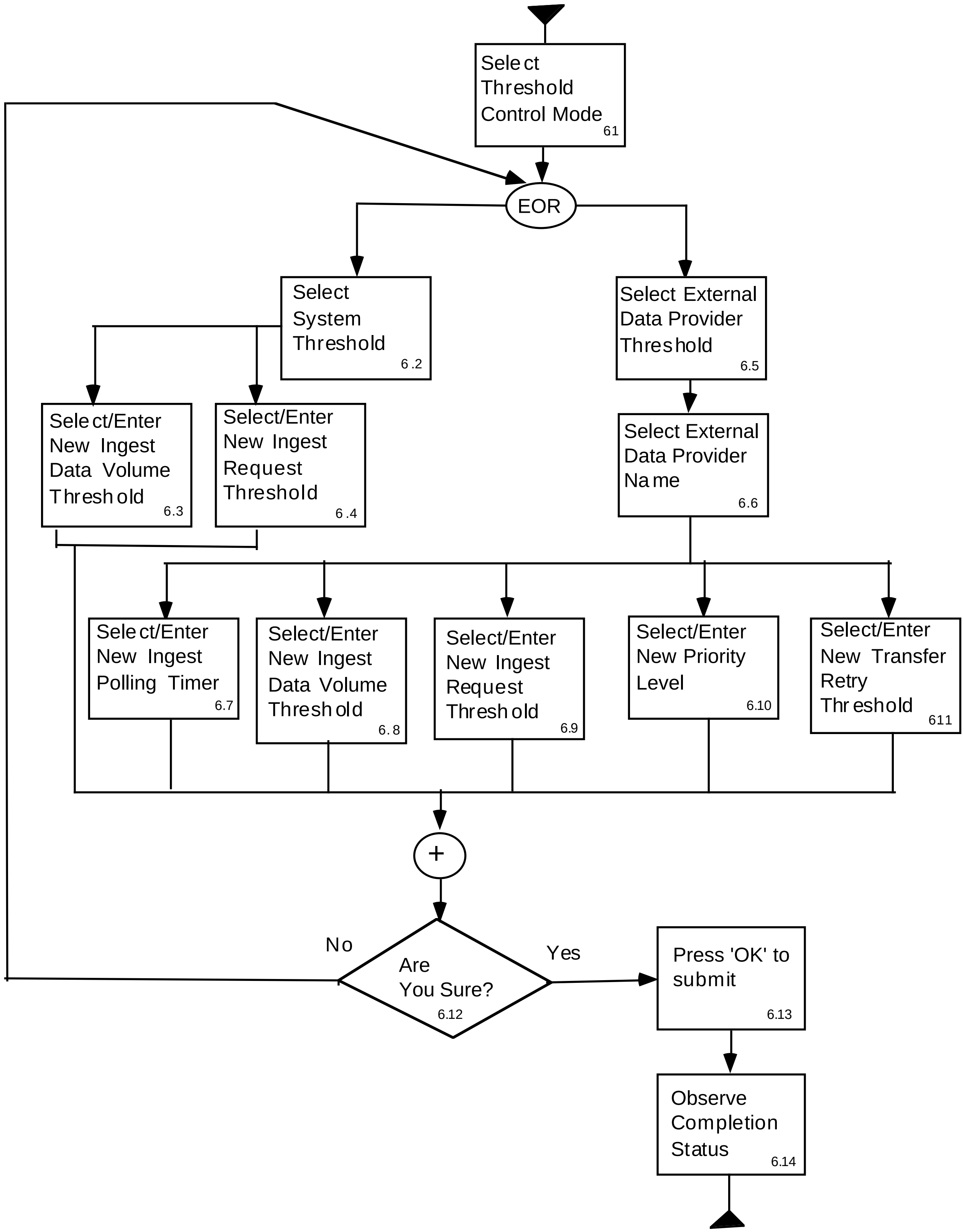


Figure C-2. Graphical Function Flow of the Ingest Threshold Control Function

Tools to Standardize the Presentation of the Operator/User Workflows

The graphical workflow diagram uses a set of tools or conventions that standardize the presentation of operator/user actions in accomplishing functions. As shown in Figure C-2, rectangular boxes are used to represent observable actions performed by operators/users in human-computer transactions. Likewise, diamond shapes are used to describe decisions made operators in these workflows. Since decision making is a purely mental activity and is generally unobservable, its occurrence must be inferred by the analyst, usually from observing the actions that an operator/user performs after making a decision.

Human-computer transactions in today's information technology systems are routinely of a branching, looping, or recursive nature. The workflow analysis depicts these branches, loops, and recursions in human-computer transactions through the application of the following tools:

a. vectors (arrows) that define the directionality of operator/user actions in a workflow

b. Boolean logic symbols (as shown in Table C-2) which define branches and/or networks of branches among which operators/users must choose the path through which they accomplish the specified function

c. looping vectors (arrows) that permit an operator/user to loop or recurse back through a previous set of operator/user actions (or branches of actions).

Table C-2. Boolean Symbols Used in Defining Workflow Branches

Symbol	Boolean Meaning
+	AND
|	OR
EOR	EXCLUSIVE OR

Basic Approach to Writing Operator/User Action or Decision Statements

The text placed inside each rectangular box or diamond shape identifies a discrete operator/user action. For maximum clarity, these actions should be described using the convention of a standard sentence. There is one convention for operator/user actions and a second convention for operator/user decisions. An action-oriented sentence is written in the following format:

Subject (noun):	No subject is recorded. The assumed subject is an operator/user.
Action Verb:	Select one of the verbs listed in Table C-3.
Object/modifiers:	Record short concise description of the object/computer tool that is manipulated by the action verb.

Table C-3. Action Verbs for Use in Performing Workflow Analysis: Motoric Action Verbs

Action Verb	Definition
Activate	To perform a control action, causing a device to become active
Close	To terminate a computer program or application
Deactivate	To perform a control action, causing a device to become inactive
Enter	To place a value or text string into a computer by means of an input/control device
Position	To indicate a 1, 2, or 3 dimensional coordinate
Select	Opt for or choose an entity by pointing to it
Set	To place a specific setting or reading in order to achieve a specified setting or mode

Perceptual Action Verbs

Action Verb	Definition
Identify	To recognize the nature of an object according to implicit or predetermined characteristics
Locate	To seek out and determine the site or place of an object
Observe	To attend visually to the presence or current status of an object
Search	Directed viewing for a specific class of objects
Scan	To quickly examine displays to obtain a general impression

Cognitive Action Verbs

Action Verb	Definition
Verify	To confirm or prove the truth of an assumption, condition, or state
Analyze	To examine critically
Choose	To select after consideration of alternatives
Decide	To come to a conclusion based on available information
Determine	To induce or deduce a conclusion or decision

The intent of the operator/user action sentence is to ensure that a short, concise, and complete statement of the operator/user action is described. The assumed subject saves space, since the operator/user is always the performer of the action that described in the human-computer transaction. As shown in Table C-3, action verbs are of three types, motoric, perceptual, and cognitive (mental). Selecting a pushbutton by means of positioning the mouse cursor and pressing a mouse button is an example of a motoric action. The verb 'Select' is appropriate to describe this action. Scanning the display to detect a change in one parameter, among a number of monitored parameters, is an example of a perceptual action. The action verb 'Observe' is appropriate for detailed monitoring actions, while the action verb 'Scan' may be more appropriate for more casual visual monitoring actions. Finally, deciding to take an action is an example of a cognitive (mental) operation. The action verb 'Decide' is appropriate for many decision making actions, while 'Determine' may be more appropriate for quick deductions or inferences.

Preparation of the Graphical Workflow Diagram

As operator/user action or decision statements are created, the analyst begins to identify the likely sequence of occurrence of these actions or decisions. This is an iterative process that becomes easier to accomplish with experience in preparing actual diagrams. To prepare a workflow diagram analysts must put themselves in the shoes of the most likely operator/user, one who possesses skills that are typical and representative of the average computer operator/user. Since developers are the analysts, in this case, and they frequently are highly skilled computer operators, it is often difficult perceive operations in the more restricted manner of an operator/user with average computer-use skills.

Sequences of operator/user action and decision statements are created as represented in Figure 2. Vectors (arrows) are used to connect these statements to one another in a manner that makes sense to the analyst, as the more likely and intuitive way in which operators/users are expected to conduct human-computer transactions with the application. As described above, a series of tools and conventions are available to the analyst to represent the branching, looping, and recursion logic that characterizes human interactions with computer applications. These can be created in the workflow diagram as the diagram evolves through iterations and greater insights into the process present themselves to the analyst. The analyst should use care to identify the Boolean logic or conditions that guide an operator in pursuit of a given sequence of actions or decisions. One final graphic convention is a small filled triangle symbol that is used to identify the start (triangle vertex down) and stop (triangle vertex up) point for each function sequence.

To facilitate the use and review of the collection of graphical workflow diagrams, the analyst should employ a numeric coding scheme which pegs an action or decision statement (a given human-machine transaction) to a specific location in the sequence of actions/decisions that define a workflow. The numbering assists the review and assessment of the workflow by others not involved in its preparation.

When complete a graphical workflow diagram will permit the reader to proceed through a sequence of operator/user actions and decisions used to accomplish specified functions. The accomplishment of operator/user functions may (and frequently will) proceed through a series of branching or looping sequence of actions or decisions. This reflects the iterative and recursive manner in which human-computer transactions are conducted. As stated, the workflows provide a representation of the activities that the GUI(s) must support. These graphical workflow diagrams are not intended to represent operator/user jobs, nor to constrain the manner in which they perform their assigned tasks, nor provide full detail on every possible action the operator/user may be required to take regarding a process. Figure C-3 illustrates another example of a workflow diagram for the Operator Media Ingest function that is identified in Table C-1.

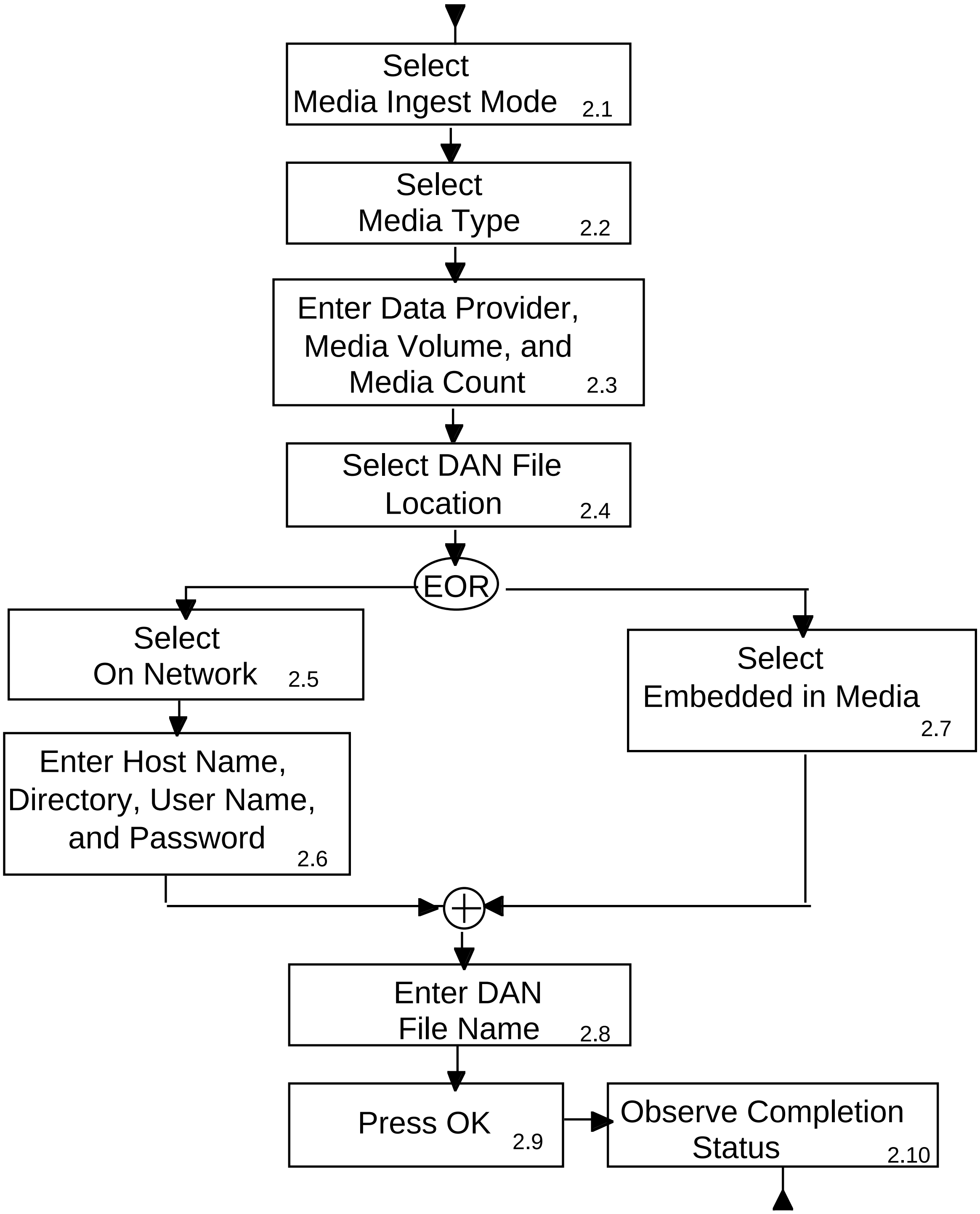


Figure C-3. Sample of a Workflow Diagram for Operator Media Ingest

Step 3: Screen Layout Drawing Packages

The purpose of this step is to define the layout of GUI screens to support the operator/user interactions and transactions identified in the workflow analysis (Step 2). The *ECS User Interface Style Guide* serves as the source of guidance to support the layout of screen elements for the design of the screens. The formality of this step may vary depending on the development team. If the designer also serves as the implementer, using the software tools to create the screens, this step may be quite informal. For example, the screen drawings can be simple pencil sketches to lay out the initial concepts, which allow the developer to move quickly to the next step. For some teams, however, the effort may be divided, with designers working to prepare the screen layouts that are used by different personnel who apply the software tools to program the screen implementations. In this case, it may be helpful to use a graphics software application to create clear drawing packages of the screen layouts to facilitate communication of the concepts from the designers to the programmers. Figure C-4 shows a screen drawing for a screen concept based on the workflow analysis represented in Figure C-3.

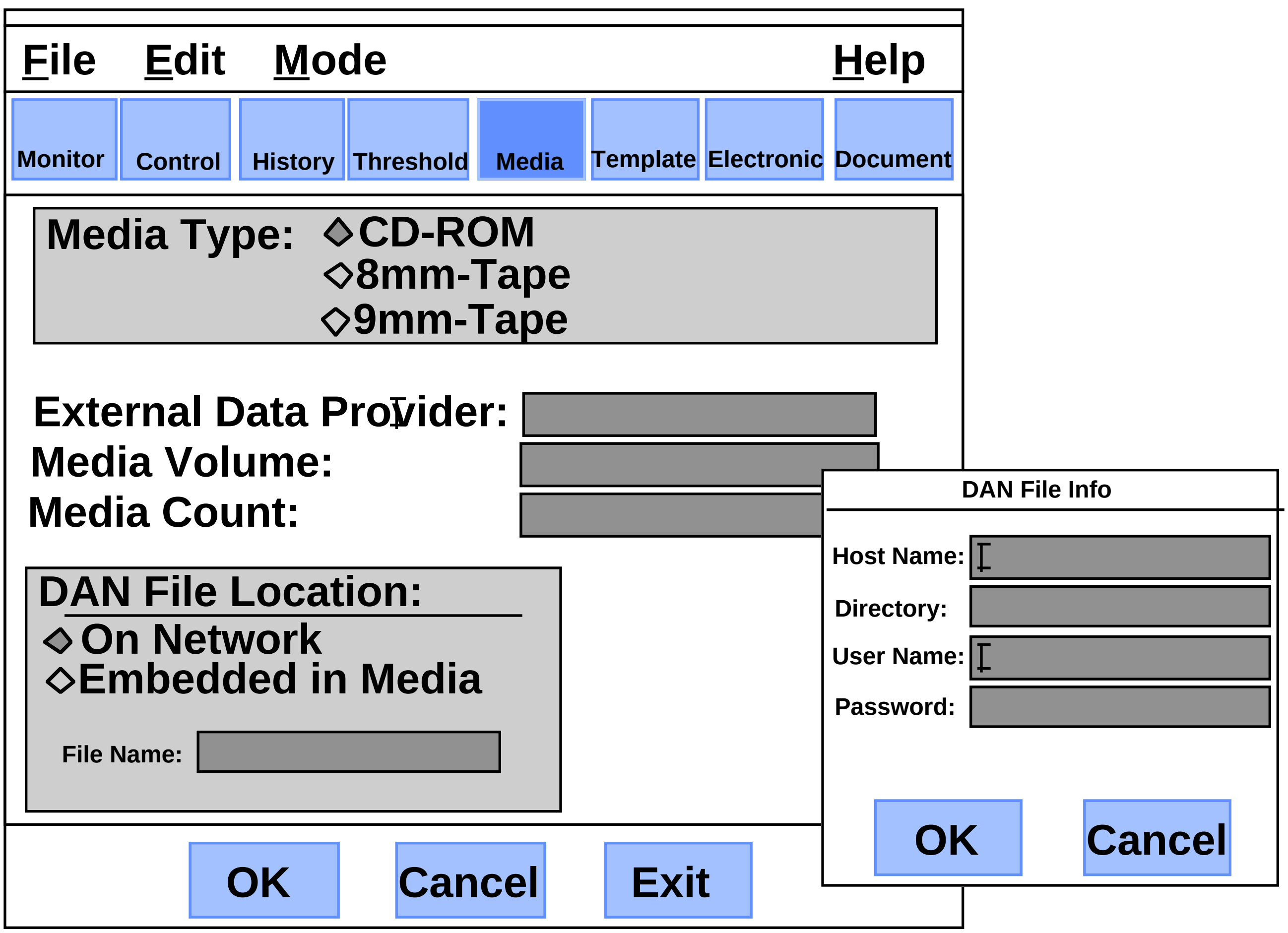


Figure C-4. Sample of a Screen Diagram for Operator Media Ingest

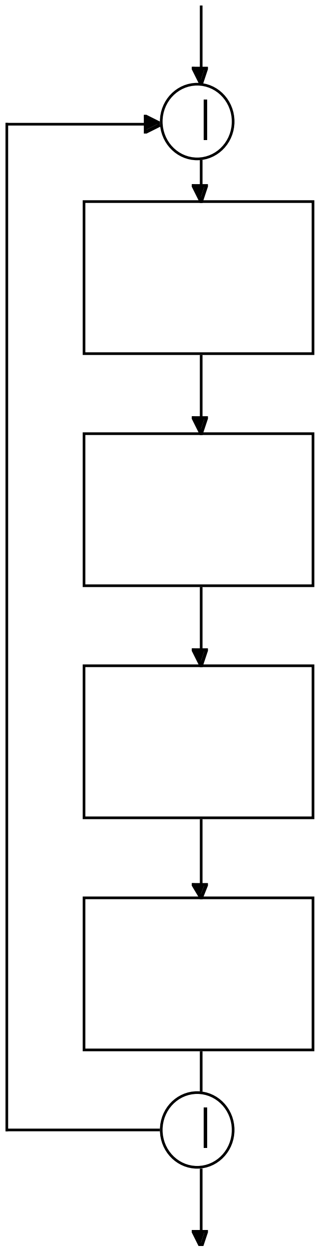
Step 4: GUI Screen Development

Preparation of GUI screens will proceed through distinct stages of development, including a rapid prototyping stage that permits human factors assessments of each proposed GUI screen to be reviewed. This step is described in the yet to be published *Motif Developer's Guide*. Most Motif GUI development will be performed using *Builder Xcessory, version 3.5.1* as the GUI builder tool. Other tools and/or Motif widgets should be approved in accordance with guidelines contained in the *Motif Developer's Guide*. The guide also prescribes guidelines for preparation of GUI code including items such as x-app defaults file that permit the tailoring of GUIs to local use by DAAC personnel.

Readers of this guideline for the HMI methodology are referred to the *Motif Developer's Guide* for information on Motif implementation standards.

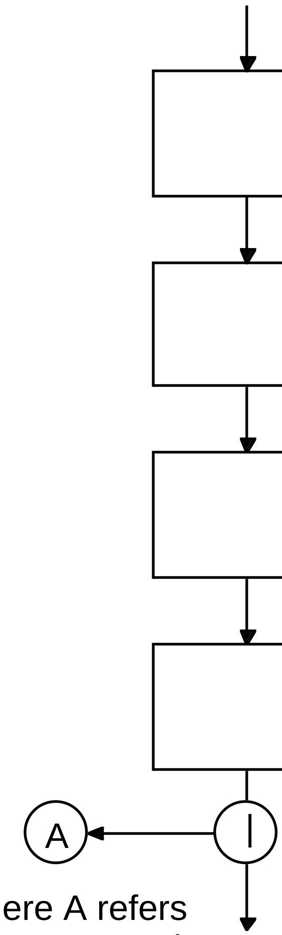
Branching Techniques for Preparing Graphical Workflows.

The following five figures present branching techniques that may be useful in preparing graphical workflows. Some of these techniques are useful for linking workflows across multiple pages of an analysis or across multiple (but interconnected) workflows.

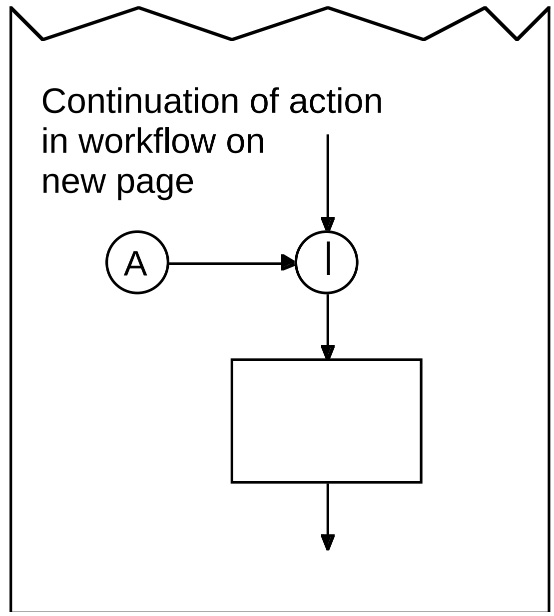


Looping structure when loop is in-line for a single workflow on the same page

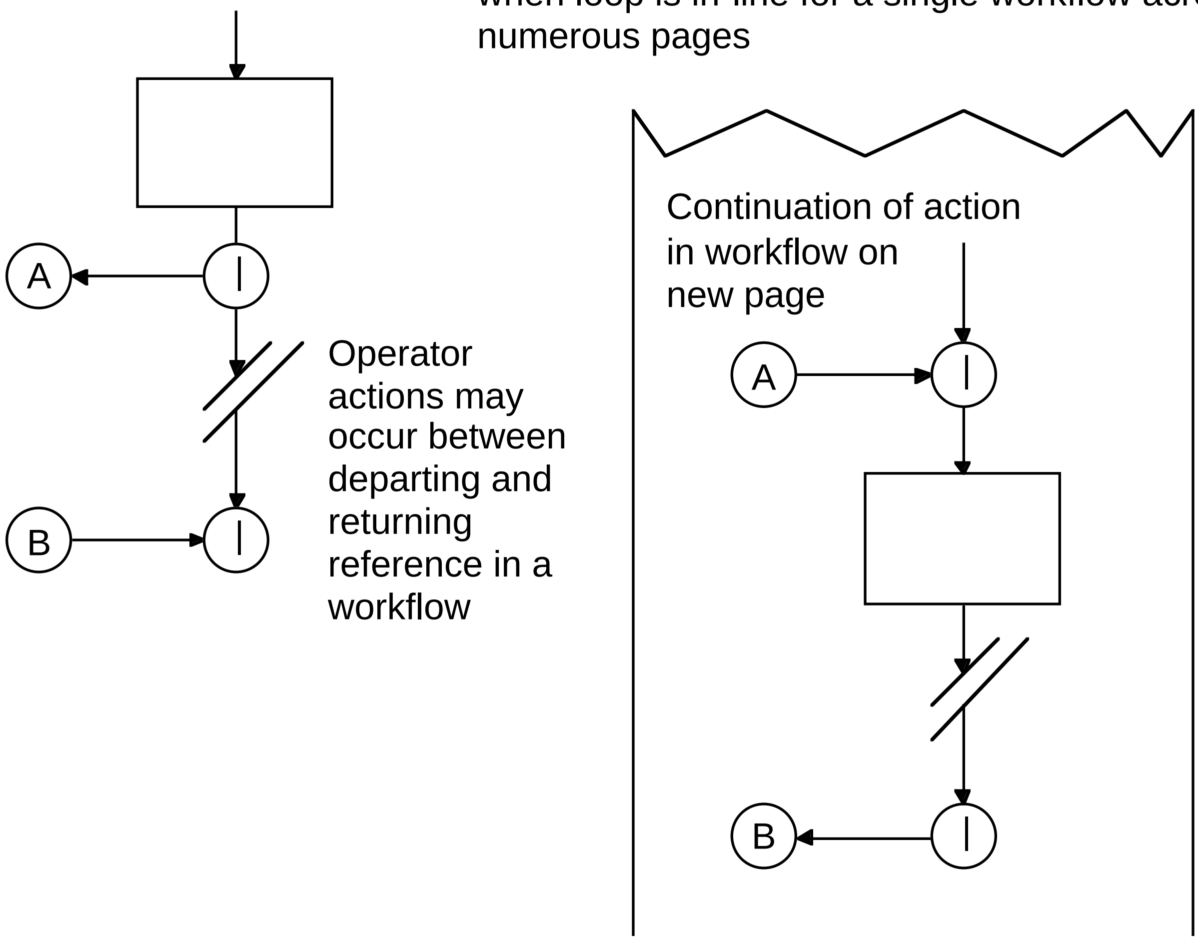
Looping structure when loop is in-line for a single workflow on separate pages

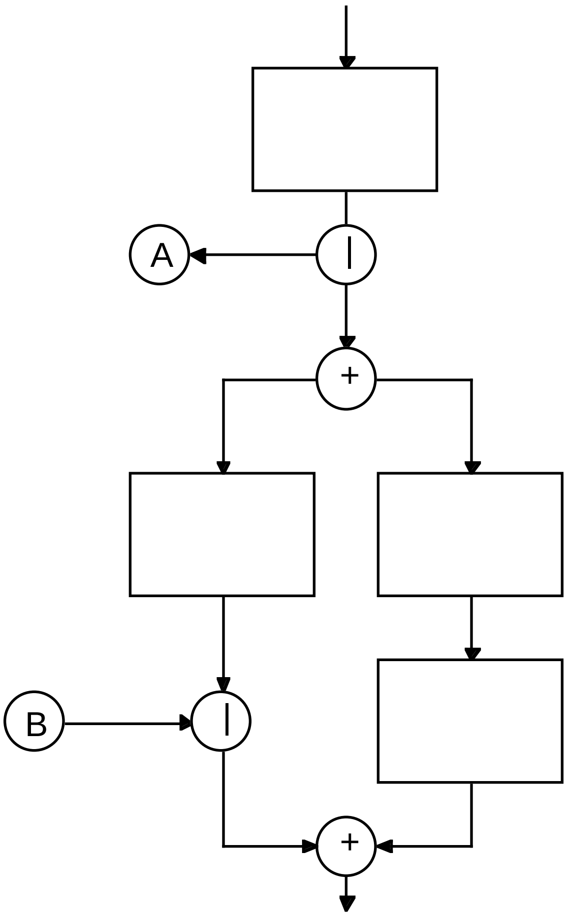


where A refers
to an entry point
on a new page of the workflow

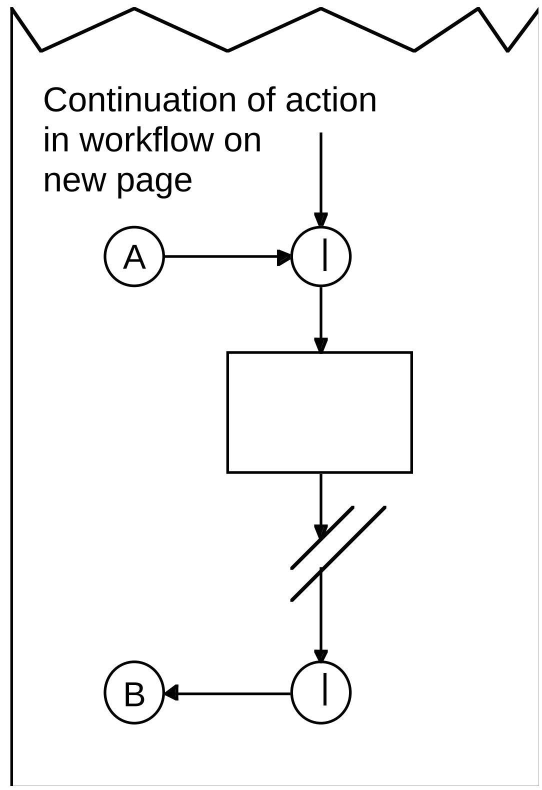


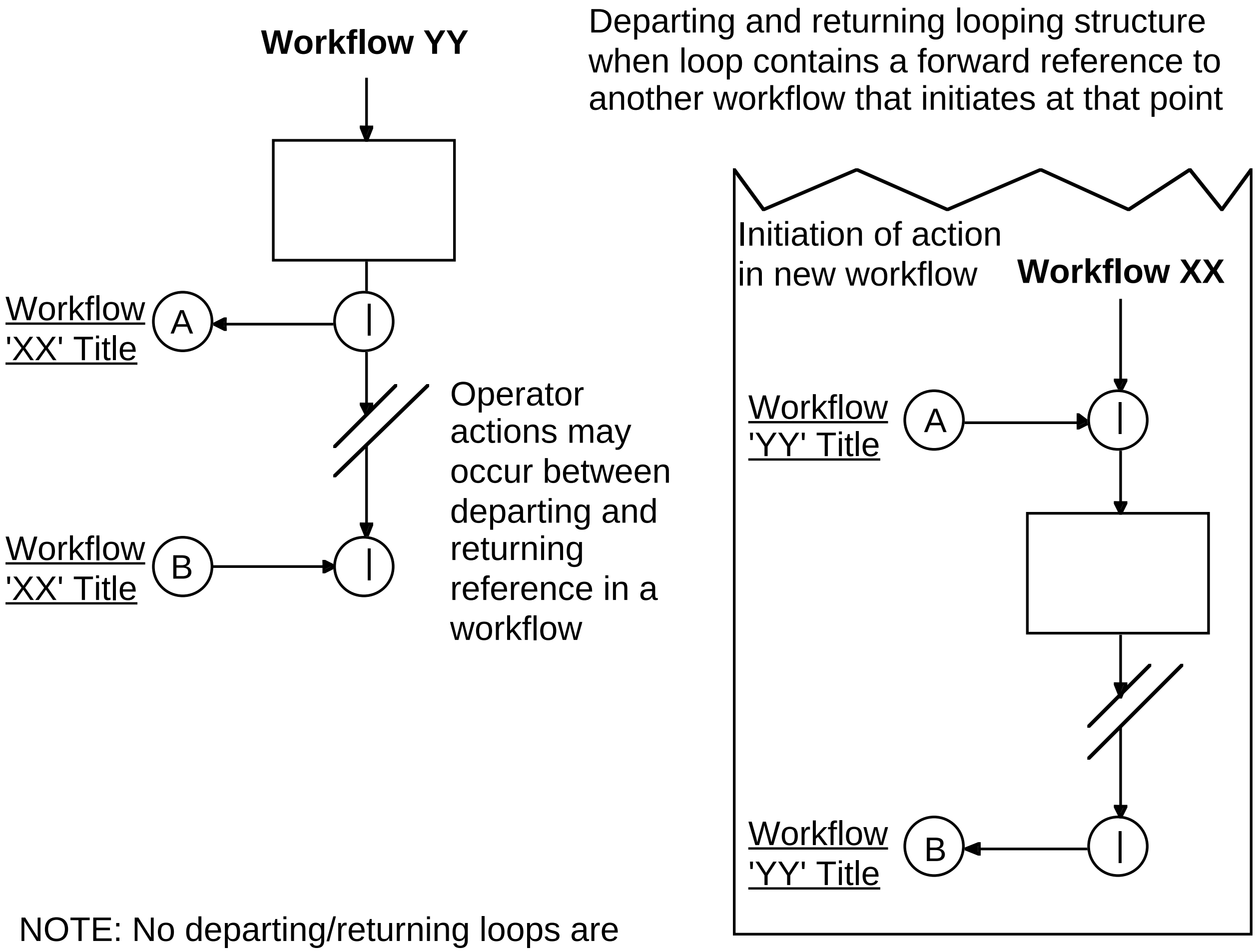
Departing and returning looping structure when loop is in-line for a single workflow across numerous pages





Departing and returning looping structure when returning loop is inside a branching workflow structure





NOTE: No departing/returning loops are allowed to depart/return inside a Boolean OR (|) or an EOR branch.

Abbreviations and Acronyms

API	Application Programming Interface
ASTER	Advanced Spaceborne Thermal Emission and Reflection Radiometer
DAR	Data Acquisition Request
DOF	Distributed Object Framework
ECS	EOSDIS Core System
EDC	Eros Data Center
ESST	Earth Science Search Tool
FROM	Functional Requirements for Mission Operations
GUI	Graphical User Interfaces
GDS	Ground Data System
IDG	Infrastructure Development Group
ICD	Interface Control Document